visual engineering
LIGHTWARE

25G-FR160

User's Manual

SAFETY INSTRUCTIONS

Class I apparatus construction. This equipment must be used with a main power system with a protective earth connection. The third (earth) pin is a safety feature, do not bypass or disable it.

This equipment should be operated only from the power source indicated on the product.

To disconnect the equipment safely from power, remove the power cord from the rear of the equipment, or from the power source. The MAINS plug is used as the disconnect device, the disconnect device shall remain readily operable.

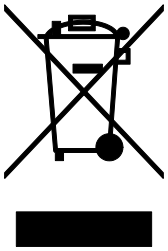
The equipment is covered by screw-fixed metal plates. Removing the covers may expose dangerous voltages.

The apparatus shall not be exposed to dripping or splashing and that no objects filled with liquids, such as vases, shall be placed on the apparatus.

The apparatus must be safely connected to multimedia systems. Follow instructions described in this manual.

WEEE (Waste Electrical & Electronic Equipment)

Correct Disposal of This Product



This marking shown on the product or its literature, indicates that it should not be disposed with other household wastes at the end of its working life. To prevent possible harm to the environment or human health from uncontrolled waste disposal, please separate this from other types of wastes and recycle it responsibly to promote the sustainable reuse of material resources.

Household users should contact either the retailer where they purchased this product, or their local government office, for details of where and how they can take this item for environmentally safe recycling.

Business users should contact their supplier and check the terms and conditions of the purchase contract. This product should not be mixed with other commercial wastes for disposal.

CE

DECLARATION OF CONFORMITY

We,

Lightware Kft. 1071 Budapest Peterdy str. 15 HUNGARY

as manufacturer declare, that the products

25G-FR160
(Hybrid modular matrix)

in accordance with the EMC Directive 2004/108/EC and the Low Voltage Directive 2006/95/EEC are in conformity with the following standards:

EMI/EMC ..... EN 55103-1 E3, EN 55103-2

Safety ..... EN 60065 Class I

Date: 01 November 2013

Name: Gergely Vida (Managing Director)

Signed: Vida A. Gergely

Table of contents

1. INTRODUCTION 9

1.1. DESCRIPTION 9
1.2. FEATURES 9
1.3. 25G HYBRID CONCEPT 10
1.4. APPLICATIONS 10

2. CONTROLS AND CONNECTIONS 11

2.1. FRONT VIEW 11
2.2. INTERNAL VIEW 13
2.3. REAR VIEW 14
2.4. CONTROL INTERFACE 15
2.5. PORT NUMBERING 16
2.6. SINGLE BOARD COMPUTER (SBC) 17
2.7. CPU BOARD (25G-CPU) 17
2.8. I/O BOARDS 18
2.8.1. I/O board designations 18
2.8.2. Status LEDs 18
2.8.3. Input boards 18
2.8.4. Output boards 19
2.9. ELECTRICAL CONNECTIONS 19
2.9.1. Fiber optical connectors 19
2.9.2. HDMI inputs and outputs 20
2.9.3. DVI inputs and outputs 20

3. INSTALLATION 21

3.1. BEFORE FIRST USE 21
3.2. UNPACKING AND MOVING OF THE UNIT 21
3.3. HANDLING OF THE BOARDS 22
3.4. POWER SUPPLIES 22

4. 25G HYBRID CONTROL SOFTWARE 23

4.1. SOFTWARE SETUP 23
4.1.1. Local control – running on the built-in PC 23
4.1.2. Remote control – running on an external PC 23
4.2. CONTROL SOFTWARE STARTUP 23
4.3. WINDOW PROPERTIES 25
4.4. CROSSPOINT MENU 26
4.4.1. Crosspoint menu – layout 26
4.4.2. Buttons and symbols 27
4.4.3. Selection panel 28
4.4.4. View mode 28
4.4.5. Take and Autotake 28
4.4.6. Source switch mode 28
4.4.7. Destination switch mode 29
4.4.8. Muting input/output port(s) 29
4.4.9. Locking input/output ports 29
4.4.10. Synchronizing 30
4.4.11. Parameters panel 31
4.5. EDID MENU 31
4.5.1. About EDID memory 31
4.5.2. EDID menu layout 32
4.5.3. Changing the emulated EDID at one input 32
4.5.4. Changing the emulated EDID at more inputs 33
4.5.5. Learning EDID 33
4.5.6. Exporting EDIDs 33
4.5.7. Importing EDIDs 33
4.5.8. Deleting EDIDs 33
4.6. ROOM EDIT 34

4.6.1. Room Edit window layout 35
4.6.2. Creating a room 35
4.6.3. Editing a room 36
4.6.4. Deleting a room 36
4.6.5. Change the name and icon of a room 36
4.7. SETTINGS 37
4.7.1. Communication Settings 37
4.7.2. User management 37
4.7.3. Control Settings 38
4.7.4. Health Status 40
4.7.5. Log Retriever 40
4.7.6. License Manager 41
4.7.7. Report tab 41
4.7.8. Advanced Viewer 42

5. PROGRAMMER'S REFERENCE 43

5.1. LW3 PROTOCOL – OVERVIEW 43
5.1.1. Elements of tree structure 43
5.1.2. Escaping 45
5.1.3. Error messages 45
5.1.4. Prefix summary 46
5.2. COMMANDS 46
5.2.1. Get all children of a node 47
5.2.2. Get all properties and methods of a node 47
5.2.3. Get all child nodes, properties and methods of a node 48
5.2.4. Set command 48
5.2.5. Invocation 49
5.2.6. Subscription 50
5.2.7. Notifications about the changes of the node structure 51
5.2.8. Notifications about the changes of the properties 51
5.2.9. Signature 52
5.2.10. Manual 53
5.2.11. Formal definitions 53
5.2.12. Source and destination identifiers 53
5.3. EXAMPLES AND SAMPLES 54
5.3.1. Switching Methods 54
5.3.2. Modify multiple connections 56
5.3.3. Disconnect multiple sources from multiple destinations 57
5.3.4. View connections on all destinations 58
5.3.5. View connections on all sources 58
5.4. PORT STATUS 59
5.4.1. View Port Status on all destinations 59
5.4.2. View Port Status on all sources 59
5.5. LW3 TREE STRUCTURE AND REFERENCE 60
5.5.1. / (root element) 60
5.5.2. /MANAGEMENT/ 60
5.5.3. /MANAGEMENT/CONFIG/ 60
5.5.4. /MANAGEMENT/CONTROL/ 60
5.5.5. /MANAGEMENT/CONTROL/<name_of_controller>/ 61
5.5.6. /MANAGEMENT/CONTROL/SERIAL/ 62
5.5.7. /MANAGEMENT/DATETIME/ 62
5.5.8. /MANAGEMENT/EDID/ 63
5.5.9. /MANAGEMENT/EDID/D/ 63
5.5.10. /MANAGEMENT/EDID/D/D<EDID_number>/ 63
5.5.11. /MANAGEMENT/EDID/F/ 64
5.5.12. /MANAGEMENT/EDID/F/F<EDID_number>/ 65
5.5.13. /MANAGEMENT/EDID/U/ 65
5.5.14. /MANAGEMENT/EDID/U/U<EDID_number>/ 65
5.5.15. /MANAGEMENT/GENLOCK/ 65
5.5.16. /MANAGEMENT/INFRA/ 65
5.5.17. /MANAGEMENT/INFRA/DATABASE/ 66

5.5.18. /MANAGEMENT/INFRA/DATABASE/<manufacturer>/ 66
5.5.19. /MANAGEMENT/INFRA/DATABASE/<manufacturer>/<device>/ 66
5.5.20. /MANAGEMENT/LAN/ 67
5.5.21. /MANAGEMENT/LAN/CPU<number>/ 67
5.5.22. /MANAGEMENT/LAN/SBC/ 68
5.5.23. /MANAGEMENT/LICENSE/ 68
5.5.24. /MANAGEMENT/LICENSE/<license_code>/ 68
5.5.25. /MANAGEMENT/LOG/ 69
5.5.26. /MANAGEMENT/SYSTEM/ 70
5.5.27. /MANAGEMENT/SYSTEM/<board_type><id>/ 73
5.5.28. /MANAGEMENT/SYSTEM/<board_type><id>/HEALTH/ 73
5.5.29. /MANAGEMENT/SYSTEM/<board_type><id>/UID/ 73
5.5.30. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/ 74
5.5.31. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/HEALTH/ 74
5.5.32. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/MNT/ 74
5.5.33. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/UID/ 74
5.5.34. /MANAGEMENT/SYSTEM/CCB<id>/ 74
5.5.35. /MANAGEMENT/SYSTEM/CCB<id>/UID/ 75
5.5.36. /MANAGEMENT/SYSTEM/CPUB<id>/ 75
5.5.37. /MANAGEMENT/SYSTEM/CPUB<id>/HEALTH/ 75
5.5.38. /MANAGEMENT/SYSTEM/CPUB<id>/UID/ 76
5.5.39. /MANAGEMENT/SYSTEM/SB/ 76
5.5.40. /MANAGEMENT/SYSTEM/SB/HEALTH/ 76
5.5.41. /MANAGEMENT/SYSTEM/SB/MNT/ 76
5.5.42. /MANAGEMENT/SYSTEM/SB/UID/ 76
5.5.43. /MANAGEMENT/USERS/ 76
5.5.44. /MANAGEMENT/USERS/<name_of_user>/ 77
5.5.45. /MANAGEMENT/USERS/root/ 78
5.5.46. /MANAGEMENT/VCP/ 78
5.5.47. /MANAGEMENT/VCP/INFRA/ 78
5.5.48. /MANAGEMENT/VCP/INFRA/V<port_number>/ 78
5.5.49. /MANAGEMENT/VCP/INFRA/V<port_number>/DATABASE/ 79
5.5.50. /MANAGEMENT/VCP/KVM/ 79
5.5.51. /MANAGEMENT/VCP/KVM/V<port_number>/ 80
5.5.52. /MANAGEMENT/VCP/RS232/ 80
5.5.53. /MANAGEMENT/VCP/RS232/V<port_number>/ 80
5.5.54. /ROOM/ 80
5.5.55. /ROOM/<name_of_room>/ 81
5.5.56. /ROOM/<name_of_room>/EDID/ 82
5.5.57. /ROOM/<name_of_room>/MONITOR/ 82
5.5.58. /ROOM/<name_of_room>/NAMES/ 83
5.5.59. /ROOM/<name_of_room>/NAMES/<name_of_layer>/ 83
5.5.60. /ROOM/<name_of_room>/PORTS/ 83
5.5.61. /ROOM/<name_of_room>/PORTS/<name_of_layer>/ 84
5.5.62. /ROOM/<name_of_room>/PORTS/<name_of_layer>/I<logical_port_number>/ ... 84
5.5.63. /ROOM/<name_of_room>/PORTS/<name_of_layer>/O<logical_port_number>/.. 84
5.5.64. /ROOM/<name_of_room>/PORTS/<name_of_layer>/V<logical_port_number>/ .. 84
5.5.65. /ROOM/<name_of_room>/PRESET/ 84
5.5.66. /ROOM/<name_of_room>/PRESET/<name_of_preset>/ 85
5.5.67. /ROOM/<name_of_room>/SALVO/ 86
5.5.68. /ROOM/<name_of_room>/SALVO/<name_of_salvo>/ 87
5.5.69. /ROOM/<name_of_room>/SETTINGS/ 88
5.5.70. /ROOM/<name_of_room>/SETTINGS/<name_of_layer>/ 89
5.5.71. /ROOM/<name_of_room>/XP/ 90
5.5.72. /ROOM/<name_of_room>/XP/<name_of_layer>/ 91
5.5.73. /ROOM/ROUTER/ 97
5.6. 25G-8HDMI1-IB AND 25G-8DVID1-IB 97
5.6.1. Video layer 97
5.7. 25G-8HDMI1-OB AND 25G-8DVID1-OB 99
5.7.1. Video layer 99
5.8. 25G-8HDMI2-A1-IB 100

5.8.1. Main node 100
5.8.2. Parameters node 101
5.8.3. Embedded Audio node 103
5.9. LW2-COMPATIBILITY 105
5.10. LW2 STATUS COMMANDS 106
5.10.1. View product type 106
5.10.2. View serial number 106
5.11. LW2 CONTROL COMMANDS 106
5.11.1. Switch one input to one output 106
5.11.2. Switch one input to all outputs 106
5.11.3. View connection on an output 107
5.11.4. View connection on all outputs 107
5.11.5. Mute specified output 107
5.11.6. Unmute specified output 108
5.11.7. Lock specified output 108
5.11.8. Unlock specified output 108

6. SPECIFICATIONS 109

6.1. TECHNICAL DRAWINGS 111

7. WARRANTY 112

8. DOCUMENT REVISION HISTORY 112

1. Introduction

Thank you for choosing Lightware 25G hybrid modular matrix. 25G Hybrid is a complete source to display solution for all video, audio and control formats over a single CAT5/6/7 or fiber cable.

1.1. Description

25G Hybrid Signal Management introduces a completely new concept to the AV industry. The new technology allows managing, switching and extending digital and analog video, audio, Ethernet and control in a new and inventive way. Designed to deliver exceptionally high resolution image quality and 24/7 reliability.

As a comprehensive line of digital matrix switchers, transmitters and receivers, 25G Hybrid combines switching with a powerful suite of diagnostic software tools for the digital age, delivering a superior user experience.

This 25 Gigabits per second allows the transferring and switching of any existing standard video format, ensuring a reliable and future proofed platform for all signal management purposes. Supported formats include Video, Audio, Ethernet, USB KVM, IR, CEC and RS-232 control in a single chassis system.

The 25G Hybrid design includes the extenders (transmitters and receivers) infrastructure called the MODEX family (MODular EXtender). MODEX offers a full range of modular transmitters and receivers including all functions of the 25G Hybrid technology. Extension can be achieved by a single CATx cable (TPS extension) or a single fiber (OPTS/OPTM version of the MODEX), both with full functionality.

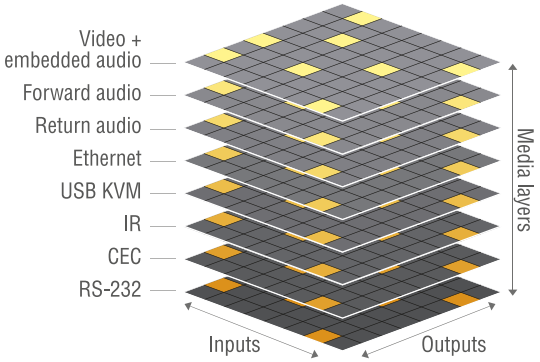
Lightware's 25G Hybrid matrix is the world's first fully compatible HDMI 1.4 matrix switcher that also provides HEC and ARC functions, supports 4K resolutions and full 3D formats. 25G Hybrid has 8 separate media layers, giving the essence to the expression: Multilayer switching.

1.2. Features

- 25 Gbit/sec per port video signal speed
- Multilayer signal management – signal switching in 3 dimensions
- Independent switching of audio and video
- USB KVM extension
- Built-in 320 port 100 Mbit Ethernet switch with 1 Gigabit uplink
- Dual redundant CPU boards for fail safe operation
- Hot swappable components, redundant power supplies – 24/7 secure operation
- RS-232 / RS-422 bidirectional transmission and control
- IR and CEC transmission
- Intuitive GUI interface for easy handling of all functions
- Room and User Management
- Front panel touch screen
- Advanced error handling and logging with time code
- Combine non-HDCP and HDCP capable I/O boards in the same frame
- TCP/IP Ethernet control (multiple connections)
- Advanced EDID Management
- Supports former LW protocols
- Barco Encore and Vista Spyder compatible

1.3. 25G Hybrid concept

One of the new investments of 25G Hybrid technology is that the signal components can be controlled separately: selecting, mixing and switching. Inside a 25G Hybrid router there are as many media layers as there are signals. This means that there are as many individual routers as there are signal formats being incorporated.



Media layers: The third switching dimension

Example 1

You have a set-top box that outputs HDMI video and audio. The same set-top box outputs the audio with a different language on its S/PDIF audio output. This box is connected to the 25G Hybrid network. Different customers can listen to the same content in different languages in different rooms.

Example 2

A media server is connected to the 25G Hybrid network inside the server room along with CD players and other equipment. The LCD displays are located in the demonstration rooms. On a certain display the picture may come from the media server, but the sound from the CD player.

1.4. Applications

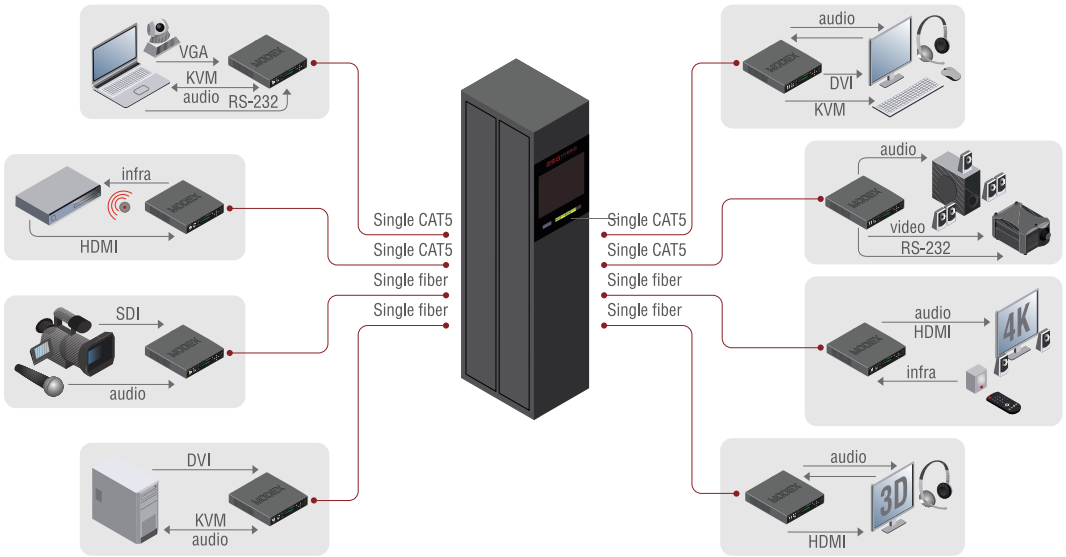
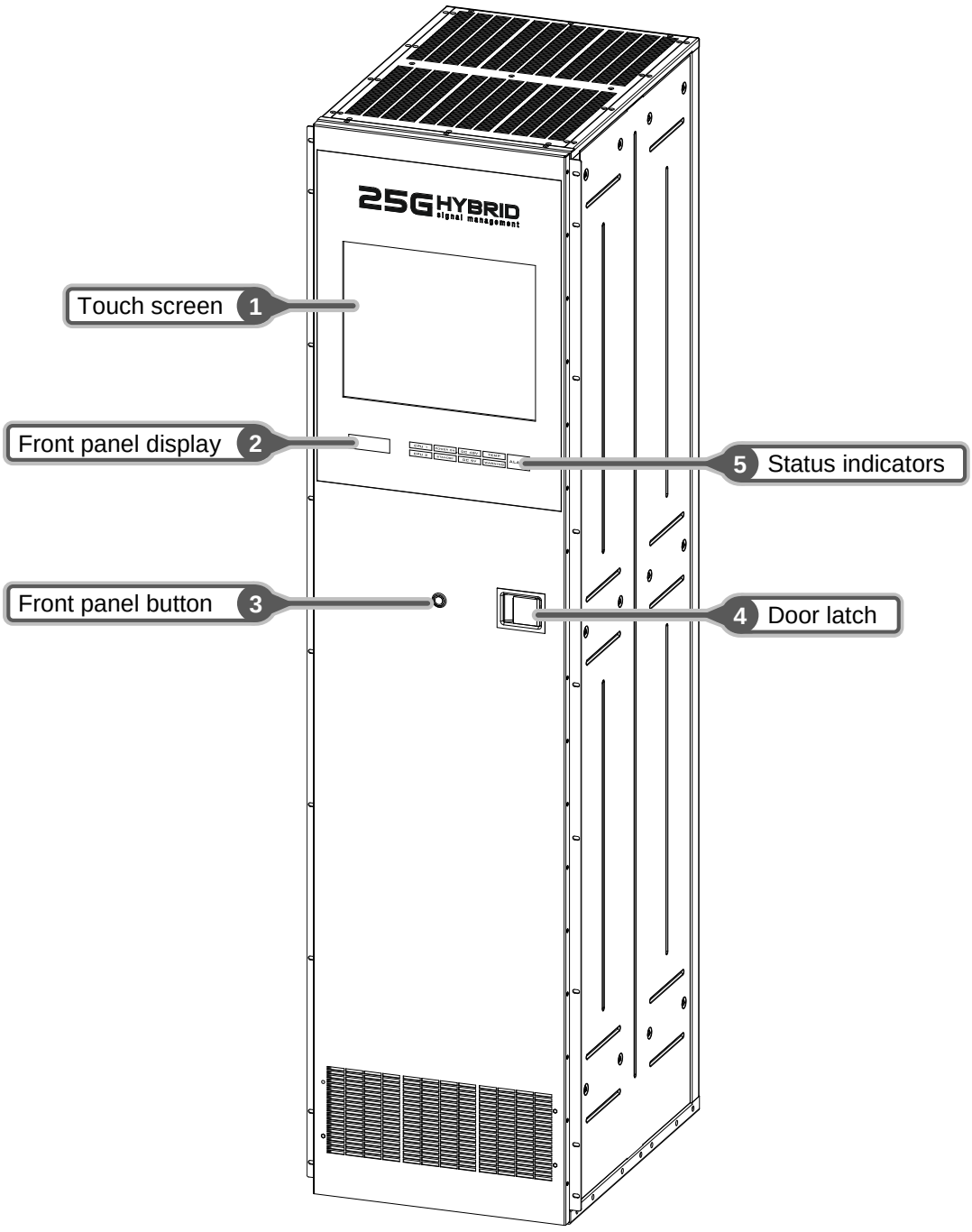


Figure 1-1. 25G Hybrid signal management

2. Controls and connections

2.1. Front view



Front view legend

1 Touch screen — Touch screen that is connected to the single board computer running the Control Software.

2 Front panel display — The following information can be shown:

- State of 25G Hybrid (Standby/Power on)
- IP address of CPU1 and CPU2 boards
- Full current of 25G Hybrid
- PSU states
- Temperature of the exhausting air
- Warning/Error events

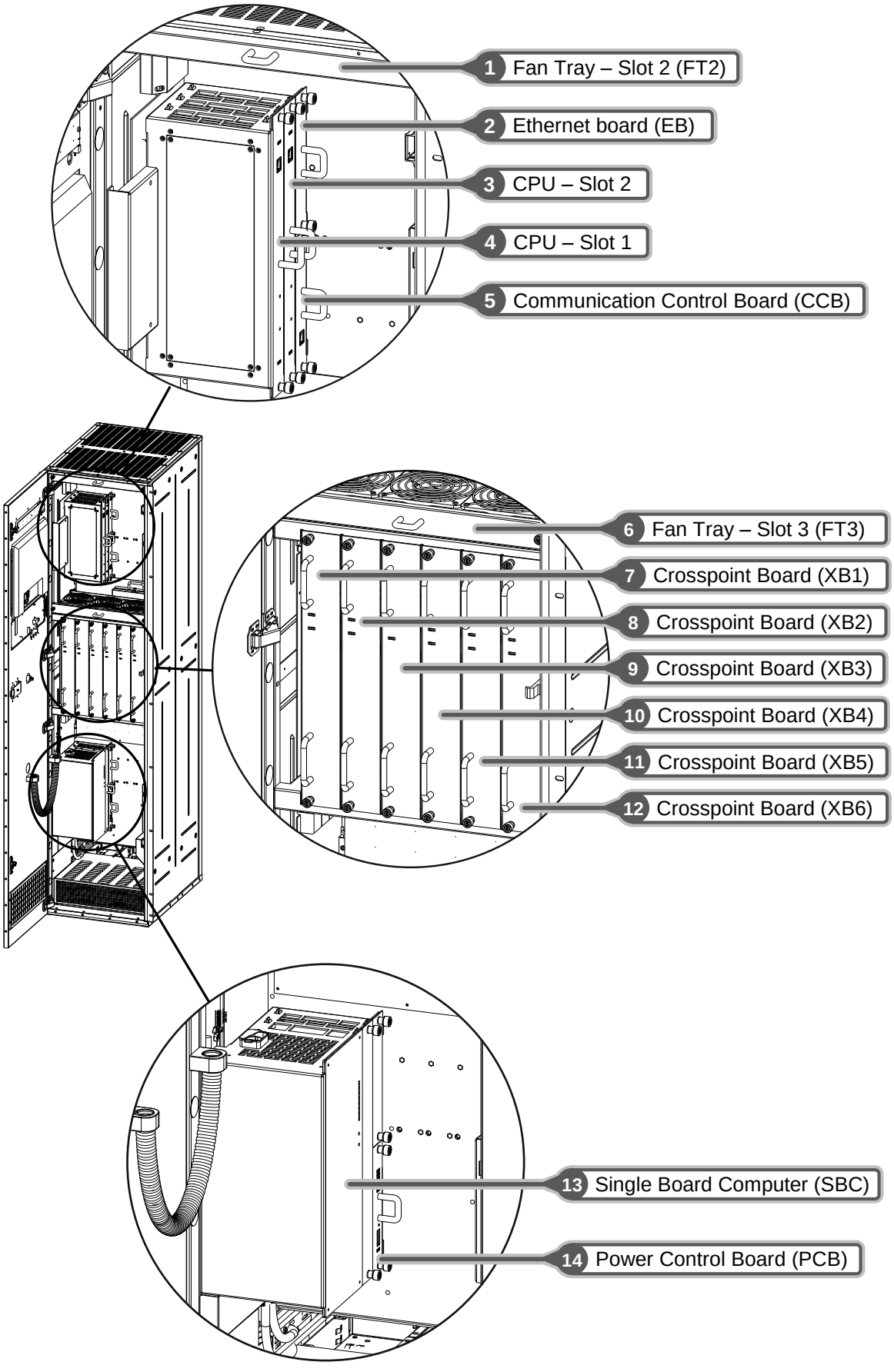
3 Front panel button — Under development

4 Door latch — Pull to open the front panel; the maximum opening angle of the door is 90°.

5 Status indicators — Displaying the following information:

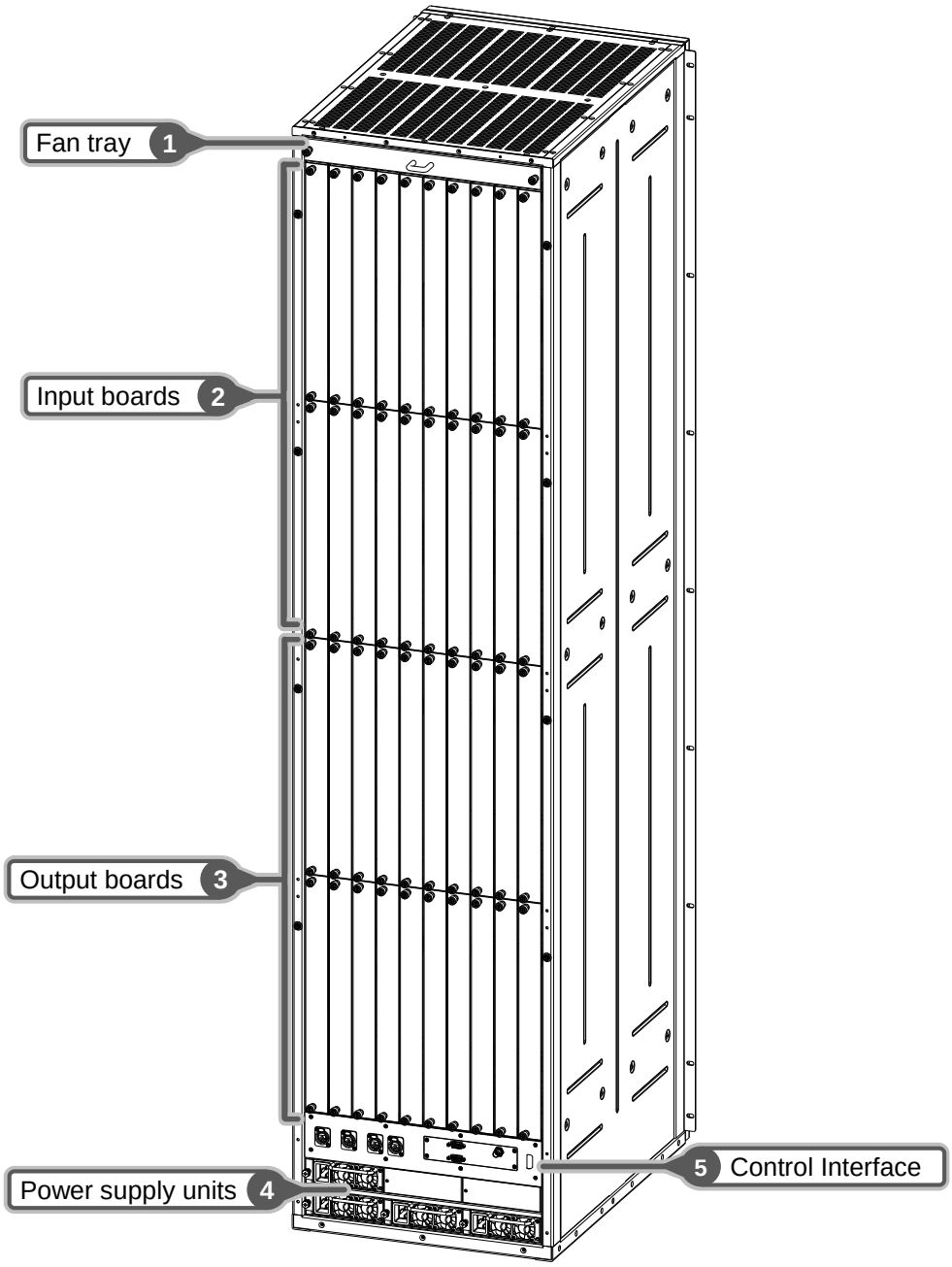
Indicator	Lamp status	Explanation
CPU1 and CPU2	dark	board not connected
	yellow	booting is in progress
	blinking yellow	firmware upgrade is in progress
	blinking green	CPU is live
	green	CPU is live and controls the matrix
	blinking red	CPU is faulty or does not operate
POWER ON	dark	the matrix is in standby mode
	green	the matrix is powered on
STANDBY	dark	the matrix is powered on
	yellow	the matrix is in standby mode
DC 48V	dark	the matrix is powered down
	green	main voltage is OK (above 46.87 V)
	blinking red	main voltage is critical (below 45.83 V)
DC 5V	green	standby voltage is OK (above 4.82 V)
	blinking red	standby voltage is critical (below 4.72 V)
TEMP	green	exhausting air temperature is below 41°C (ideal)
	blinking yellow	exhausting air temperature is between 42°C and 51°C (safe range)
	blinking red	exhausting air temperature is above 52°C (dangerous/critical)
WARNING	dark	no warning message
	blinking yellow	(not fatal) error happened during the operation
ALARM	dark	the matrix operates properly
	blinking red	fatal error happened during the operation

2.2. Internal view

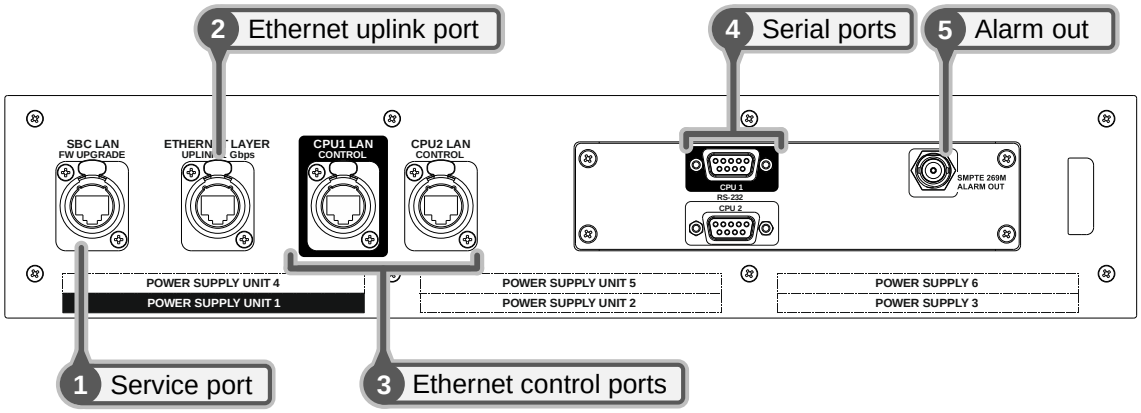

1 Fan Tray – Slot 2 (FT2)
2 Ethernet board (EB)
3 CPU – Slot 2
4 CPU – Slot 1
5 Communication Control Board (CCB)
6 Fan Tray – Slot 3 (FT3)
7 Crosspoint Board (XB1)
8 Crosspoint Board (XB2)
9 Crosspoint Board (XB3)
10 Crosspoint Board (XB4)
11 Crosspoint Board (XB5)
12 Crosspoint Board (XB6)
13 Single Board Computer (SBC)
14 Power Control Board (PCB)

2.3. Rear view

The ports of the installed boards and the control interface are available on the back of the device. Empty slots are covered by a black plate which can be fixed by the screws.



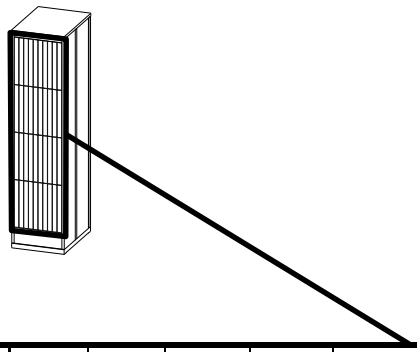
2.4. Control interface



1	**Service port**	RJ45 connector to connect a computer directly, for service and maintenance purposes.
2	**Ethernet uplink port**	RJ45 connector for connecting to Ethernet layer.
3	**Ethernet control ports**	Direct Ethernet connections separately to CPU1 and CPU2.
4	**Serial ports**	Separate RS-232 connection to CPU1 and CPU2.
5	**Alarm out**	Standard SMPTE 269M alarm output with BNC connector.

2.5. Port numbering of FR-160

The ports of the input and output boards in the 160x160 frame are numbered as follows:



Input boards	I1	I9	I17	I25	I33	I41	I49	I57	I65	I73
	I2	I10	I18	I26	I34	I42	I50	I58	I66	I74
	I3	I11	I19	I27	I35	I43	I51	I59	I67	I75
	I4	I12	I20	I28	I36	I44	I52	I60	I68	I76
	I5	I13	I21	I29	I37	I45	I53	I61	I69	I77
	I6	I14	I22	I30	I38	I46	I54	I62	I70	I78
	I7	I15	I23	I31	I39	I47	I55	I63	I71	I79
	I8	I16	I24	I32	I40	I48	I56	I64	I72	I80
	I81	I89	I97	I105	I113	I121	I129	I137	I145	I153
	I82	I90	I98	I106	I114	I122	I130	I138	I146	I154
	I83	I91	I99	I107	I115	I123	I131	I139	I147	I155
	I84	I92	I100	I108	I116	I124	I132	I140	I148	I156
	I85	I93	I101	I109	I117	I125	I133	I141	I149	I157
	I86	I94	I102	I110	I118	I126	I134	I142	I150	I158
	I87	I95	I103	I111	I119	I127	I135	I143	I151	I159
	I88	I96	I104	I112	I120	I128	I136	I144	I152	I160
Output boards	O1	O9	O17	O25	O33	O41	O49	O57	O65	O73
	O2	O10	O18	O26	O34	O42	O50	O58	O66	O74
	O3	O11	O19	O27	O35	O43	O51	O59	O67	O75
	O4	O12	O20	O28	O36	O44	O52	O60	O68	O76
	O5	O13	O21	O29	O37	O45	O53	O61	O69	O77
	O6	O14	O22	O30	O38	O46	O54	O62	O70	O78
	O7	O15	O23	O31	O39	O47	O55	O63	O71	O79
	O8	O16	O24	O32	O40	O48	O56	O64	O72	O80
	O81	O89	O97	O105	O113	O121	O129	O137	O145	O153
	O82	O90	O98	O106	O114	O122	O130	O138	O146	O154
	O83	O91	O99	O107	O115	O123	O131	O139	O147	O155
	O84	O92	O100	O108	O116	O124	O132	O140	O148	O156
	O85	O93	O101	O109	O117	O125	O133	O141	O149	O157
	O86	O94	O102	O110	O118	O126	O134	O142	O150	O158
	O87	O95	O103	O111	O119	O127	O135	O143	O151	O159
	O88	O96	O104	O112	O120	O128	O136	O144	O152	O160

2.6. Single Board Computer (SBC)

25G frames contain a built-in computer that is available via the touch screen. The PC is Linux-based and runs 25G Controller Software.

The PC was designed:

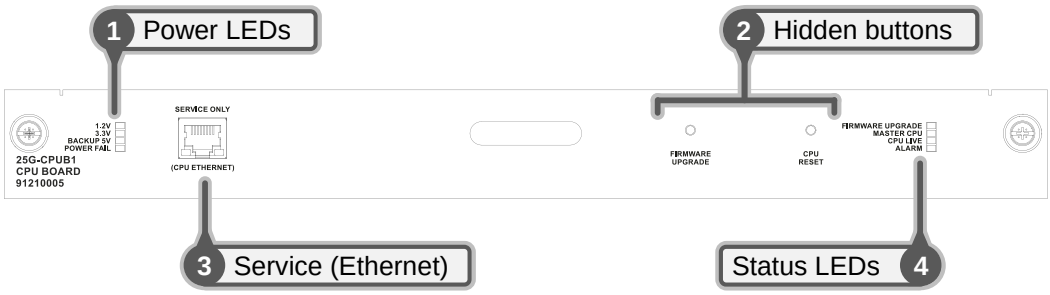
- To run 25G Controller software which makes all control functions available locally,
- To assist in firmware upgrade process.

The matrix can be controlled either from the touch screen or from a PC connected to the CPU directly. The built-in PC is connected to the CPU so, as an external computer is connected to the matrix via Ethernet or serial port.

Info: If SBC is powered off, firmware cannot be upgraded.

2.7. CPU board (25G-CPU)

25G frames can handle contain two CPU boards to ensure the constant reliability of the device. If any problem occurs, which makes the master CPU card stopping the operation, the second CPU card takes the control automatically.



1 Power LEDs Display the different DC voltage levels' status.

2 Hidden buttons Buttons for special service purposes.

3 Service (Ethernet) Ethernet connection to CPU for service purposes.

4 Status LEDs

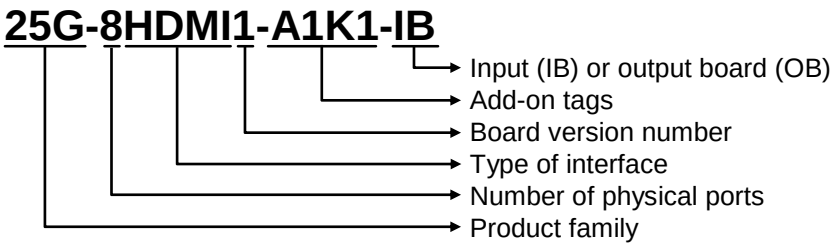
Label	LED status	Explanation
FIRMWARE UPGRADE	solid green	firmware upgrade is in progress
MASTER CPU	solid green	the board is the main controller of the matrix
CPU LIVE	blinking green	CPU is live
ALARM	dark	the CPU operates properly
	solid red	error happened during the operation

2.8. I/O Boards

Different types of input and output boards give the maximum flexibility for signal transmission. The hybrid architecture allows signal routing between boards even if they have different connectors. This way any input can be routed to any or more outputs, if the output interface is capable to transmit the signal.

2.8.1. I/O board designations

The type (code) of the boards follow the following structure:



2.8.2. Status LEDs

Label	LED status	Explanation
(no label)	not in use	-
CTRL	solid green	The board is controlled by the CPU
	blinking green	The board is not controlled by the CPU
LIVE	blinking green	The board is live (normal operation)
PWR	solid green	The board is powered on

2.8.3. Input boards

Several input interface boards are available. Each model has different capabilities and functions. The table below shows a summary of the main features.

Board type	Connectors	Capabilities
25G-8HDMI2-A1-IB	HDMI in	HDMI, HDCP, EDID, Cable EQ
	RCA	S/PDIF in
25G-8DVID2-IB	DVI-I in	HDMI, HDCP, EDID, Cable EQ
25G-8OPTS1-IB-LC	Singlemode fiber input LC / SC / ST / Neutrik OpticalCON	Audio, Video, Ethernet, USB KVM, IR, CEC, RS-232 (up to 10 km distance)
25G-8OPTS1-IB-SC		
25G-8OPTS1-IB-ST		
25G-8OPTS1-IB-NT		
25G-8OPTM1-IB-LC	Multimode fiber input LC / SC / ST / Neutrik OpticalCON	Audio, Video, Ethernet, USB KVM, IR, CEC, RS-232 (up to 300 m distance)
25G-8OPTM1-IB-SC		
25G-8OPTM1-IB-ST		
25G-8OPTM1-IB-NT		

2.8.4. Output boards

Several output interface boards are available. Each model has different capabilities and functions. The table below shows a summary of the main features.

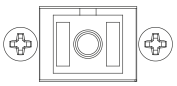
Board type	Connectors	Capabilities
25G-8HDMI2-A1-OB	HDMI out	HDMI, HDCP, EDID, Cable EQ
	RCA	S/PDIF out
25G-8HDMI2-A2-OB	HDMI out	HDMI, HDCP, EDID, Cable EQ
	RCA	S/PDIF in/out
25G-8HDMI2-A3-OB	HDMI out	HDMI, HDCP, EDID, Cable EQ
	Phoenix	Analog audio in/out
25G-8DVID2-OB	DVI-I out	HDMI, HDCP, EDID, Cable EQ
25G-8OPTS1-OB-LC	Singlemode fiber output LC / SC / ST / Neutrik OpticalCON	Audio, Video, Ethernet, USB KVM, IR, CEC, RS-232 (up to 10 km distance)
25G-8OPTS1-OB-SC		
25G-8OPTS1-OB-ST		
25G-8OPTS1-OB-NT		
25G-8OPTM1-OB-LC	Multimode fiber output LC / SC / ST / Neutrik OpticalCON	Audio, Video, Ethernet, USB KVM, IR, CEC, RS-232 (up to 300 m distance)
25G-8OPTM1-OB-SC		
25G-8OPTM1-OB-ST		
25G-8OPTM1-OB-NT		

2.9. Electrical connections

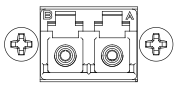
2.9.1. Fiber optical connectors

Optical boards can be assembled with several standard fiber connector types.

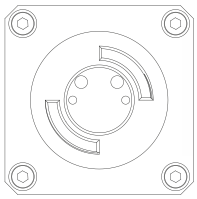
ST receptacle



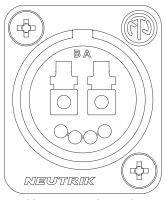
SC receptacle



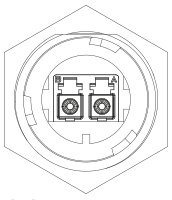
LC receptacle



EBC Junior



NT® (Neutrik OpticalCON DUO)

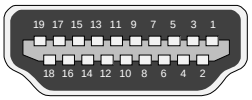


Industrial LC ODVA (Tyco)

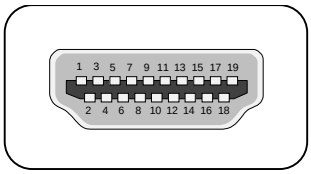
Figure 2-1. Available Fiber optical connectors

2.9.2. HDMI inputs and outputs

19-pole HDMI connectors are provided for inputs and outputs.



HDMI Type A receptacle



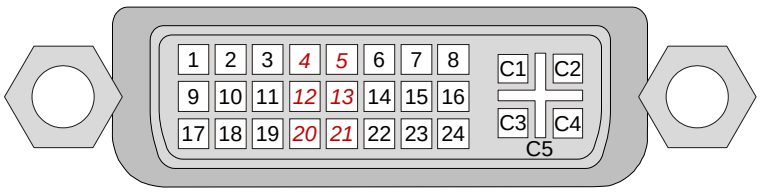
HDMI Type A Plug

Pin	Signal	Pin	Signal
1	TMDS Data2+	**11**	TMDS Clock Shield
2	TMDS Data2 Shield	**12**	TMDS Clock–
3	TMDS Data2–	**13**	CEC
4	TMDS Data1+	**14**	Reserved
5	TMDS Data1 Shield	**15**	SCL
6	TMDS Data1–	**16**	SDA
7	TMDS Data0+	**17**	DDC/CEC/HEC Ground
8	TMDS Data0 Shield	**18**	+5 V Power (max 50 mA)
9	TMDS Data0–	**19**	Hot Plug Detect
10	TMDS Clock+		

Table 2-1. HDMI connector and pin assignments

2.9.3. DVI inputs and outputs

29 pole DVI-I connectors, however internally connected pins vary by interface types. This way, user can plug in any DVI connector, but keep in mind that analog signals (such as VGA or RGBHV) are not processed. Always use high quality DVI cable for connecting sources and displays. Pay attention to the DVI cable, if dual link signal is to be sent, use only dual link capable DVI cables.



Pin	Signal	Pin	Signal	Pin	Signal
1	TMDS Data2-	**9**	TMDS Data1-	**17**	TMDS Data0-
2	TMDS Data2+	**10**	TMDS Data1+	**18**	TMDS Data0+
3	TMDS Data2/4 Shield	**11**	TMDS Data1/3 Shield	**19**	TMDS Data0/5 Shield
4	*TMDS Data4-*	**12**	*TMDS Data3-*	**20**	*TMDS Data5-*
5	*TMDS Data4+*	**13**	*TMDS Data3+*	**21**	*TMDS Data5+*
6	DDC Clock	**14**	+5V Power	**22**	TMDS Clock Shield
7	DDC Data	**15**	GND (for +5V)	**23**	TMDS Clock+
8	nc	**16**	Hot Plug Detect	**24**	TMDS Clock-
C1	nc	**C2**	nc	**C3**	nc
C4	nc	**C5**	GND		

Table 2-2. DVI-I connector pin assignments

3. Installation

3.1. Before first use

- Move the equipment carefully when it is unpacked from the packaging material; especially mind your feet and hands when standing the equipment onto the ground.
- Always use an ESD wrist strap connected to the ground when an I/O board is installed/removed into/from the frame.
- Mind the ventilators when touching the fan tray and watch your fingers.
- Never block the ventilators or the air flow!
- Keep the equipment away from rain or condensing humidity.
- Do not touch hot surfaces by hand!
- Do not disassemble or repair the power supply units!
- Always use the supplied box and packaging material if the equipment has to be shipped.
- If any component gets defective, do not start to repair it, but contact Lightware's support team: support@lightware.eu.

3.2. Unpacking and moving of the unit

Move the box as close as possible to the destination place. Follow below steps when unpacking the device:

Step 1. Unlock the bolts that fix the cover to the base plate.

Step 2. Lift the cover from the base plate by grabbing the four heels (Figure 3-1).

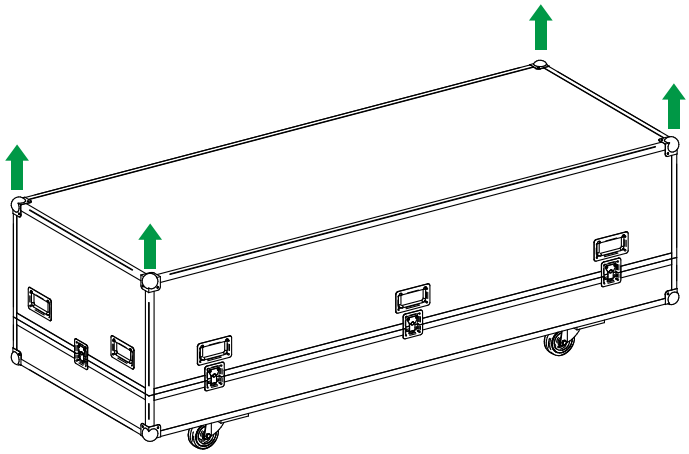


Figure 3-1. Lift the cover

Step 3. Make sure that the matrix is lifted at least by four people; grabbing at the four heels and remove it from the base plate (Figure 3-2).

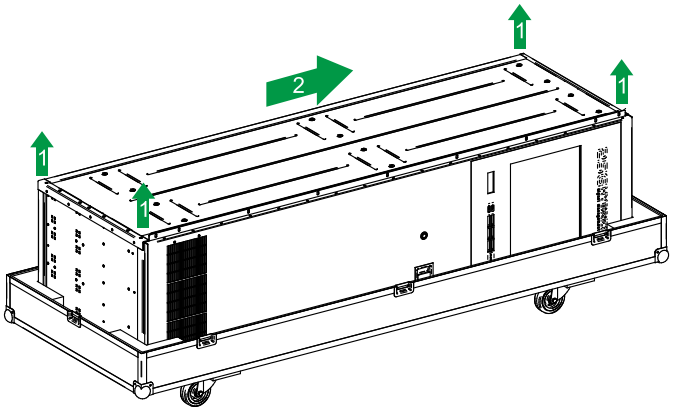


Figure 3-2. Removing from the base plate

Step 4. Pay attention to the front door: do not lift the matrix by the door or the glass! Suitable surfaces for grabbing are the rear pillars and the rack ears.

Step 5. Make the device to stand: make sure that two or three people lift at the top and another person fixes at the bottom part avoiding the device to slip (Figure 3-3).

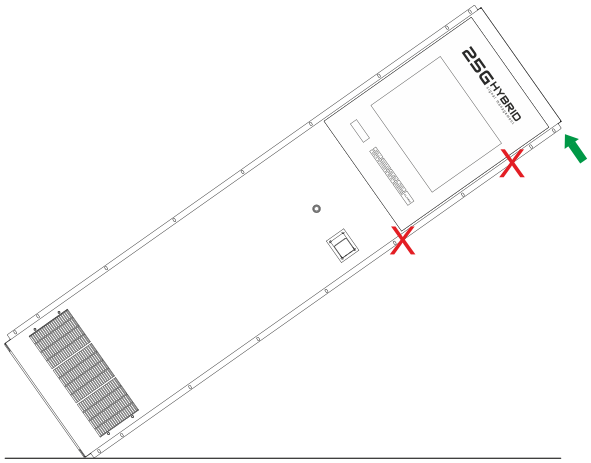


Figure 3-3. Make the device to stand

3.3. Handling of the boards

Please mind the followings when an input or output board has to be removed or installed:

- The matrix consist of numerous circuit boards assembled with sensitive electronic components. Always think of protection against ESD (protective clothing, wrist strap, etc...) when touching any board.
- Disconnect all cables from the installed board.
- Loose the fixing screws of the board and pull it carefully in the guiding rails.
- Store the unused boards in safe, ESD protected packaging.
- Fix the new board by the screws and check the LED's state to see if the install was successful.

3.4. Power supplies

The state of the power supplies can be checked on the front panel display and through the Control Software as well. The supplied two power supply units are hot swappable. If any malfunction happens with the currently used one, the device starts using the other PSU and the defective one can be replaced.

Important! *Do not open the PSU, do not start to repair if it gets defective! Always contact Lightware in case of a power supply problem.*

4. 25G Hybrid Control Software

The Software Control ("25G Controller") is different at 25G Hybrid than at other Lightware products. There are two ways to connect to the CPU of 25G:

- Connecting by RS-232 serial or Ethernet port on the Control interface and running 25G Controller on an external PC, or
- Use the Single Board Computer (the built-in PC) and use 25G Controller on the touch screen of 25G Hybrid.

Info: The surface and the functions are the same in both cases.

Info: 25G controller is Java-based, platform-independent, thus it can be run under many operating systems.

4.1. Software setup

4.1.1. Local control – running on the built-in PC

If the device is powered on 25G Controller is run automatically and can be used in a minute.

4.1.2. Remote control – running on an external PC

Step 1. Contact Lightware in order to get the latest 25G Controller Software package.

Step 2. Download and extract the package to the PC.

Step 3. Connect 25G Hybrid to the same Network where the controller PC is located – make sure they are in the same subnet. The IP address of 25G CPU1 is set to DHCP (Auto IP) as default.

Step 4. Run 'LWDeviceController.exe'.

4.2. Control Software startup

Step 1. Run 'LWDeviceController.exe'; a window will display the available devices.

Info: Be sure that the firewall is not blocking the application when running from an external PC.

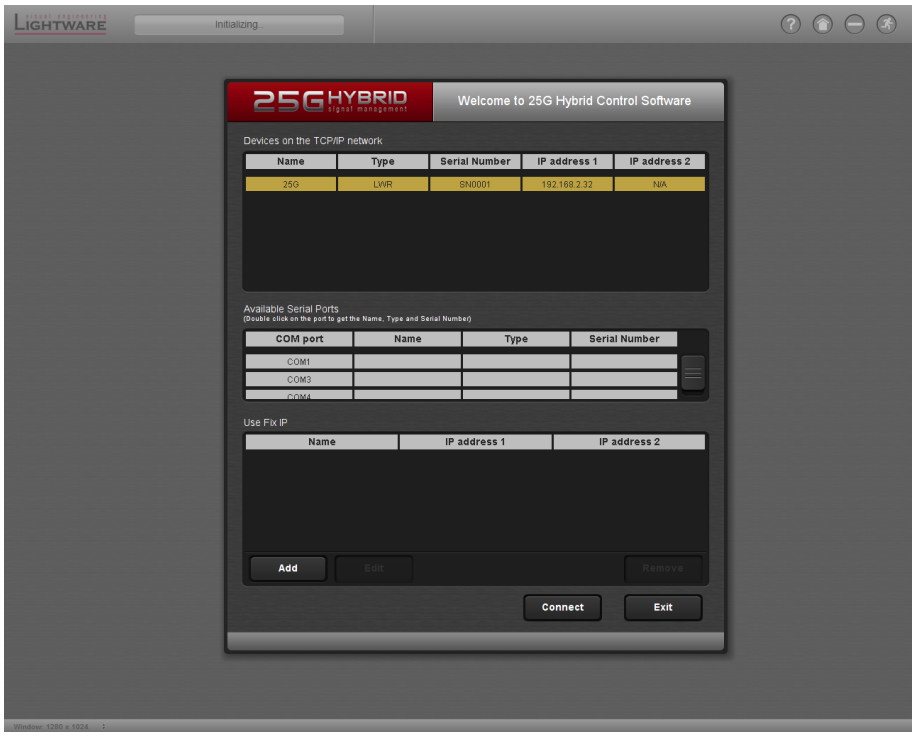


Figure 4-1. 25G Controller startup screen

Info: The whole window can be moved by grabbing the header of the small 'Welcome' window.

After the connection has been made via Ethernet, the software shows the available Lightware 25G devices in the upper list. The device name, type, serial number and IP address are displayed automatically; if CPU2 board is installed and set, its IP address is also displayed in the last column. Press the desired device, to highlight it.

You can also add devices with fix IP address that are not in above list by pressing the 'Add' button. Write a name that will identify the device, then fill the 'IP address' and 'Port number' fields – if CPU2 is installed, its data can be written in the second row. The listed devices can be modified by the 'Edit' button, or removed by the 'Remove' button.

Info: Set the port number to 6107 if LW3 protocol is used for the communication.

The list of devices connected via serial port are inquired only by double clicking the appropriate port. The device name, type and serial number are displayed; it can be highlighted with a single click.

Step 2. Select a device and press the 'Connect' button.

When the connection is established, the login screen pops up.

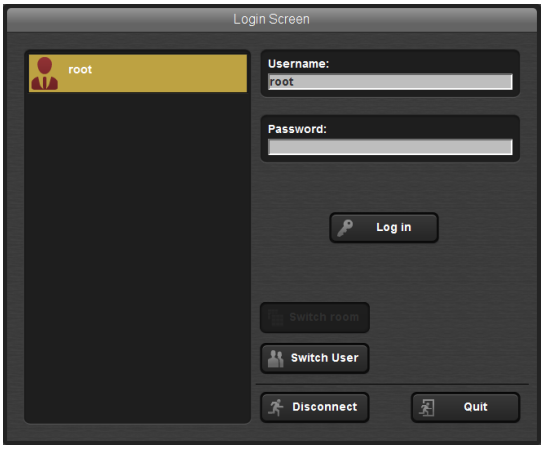


Figure 4-2. Login screen

Select a user from the list on the left side, type the password and press 'Log in'.

Default user name: root

Default password: admin

Info: The login screen can be turned OFF and set the Control Software to login with root user directly. See more information in section 4.7.2 on page 37.

Step 3. Select a room and press the 'ENTER' button.

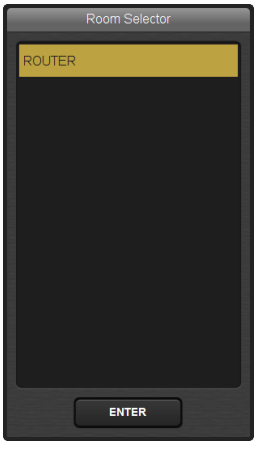
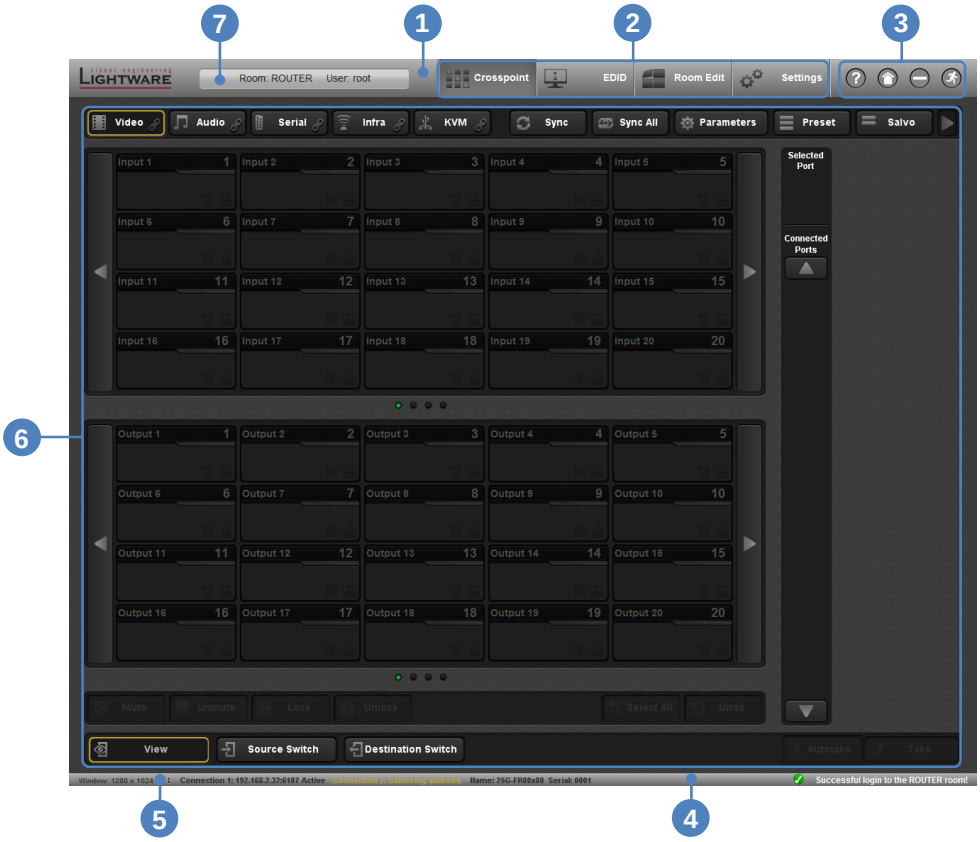


Figure 4-3. Room selector

Default room: ROUTER (contains all input/output ports of all layers)

4.3. Window properties

Layout



1 Grabbing area

The window can be moved by grabbing the indicated surface. The window is resized to full screen automatically by double-clicking on the grabbing area.

2 Main menu

Shows the available menu items which depends on the user's rights: not allowed menu will not be displayed.

3 Window icons

Displaying 'About' window.

Jump to 'Home page': default menu will be displayed (Crosspoint).

Minimize the Control Software to system tray.

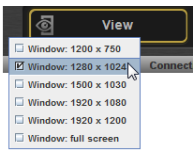
Lock screen; in the pop-up window you can switch user, switch room, disconnect from the device or quit from the program.

4 Status line

Displaying the device's name and serial number with the IP addresses of the active CPU boards.

5 Screen resolution

Select the resolution that fits the best for your screen. After selecting the desired resolution, the layout is arranged automatically.



6 Active area

Functions and information are displayed according to the selected menu item.

7 Room selector

Click on the grey area to display the room selector panel; you can switch to another defined room.

4.4. Crosspoint menu

4.4.1. Crosspoint menu – layout

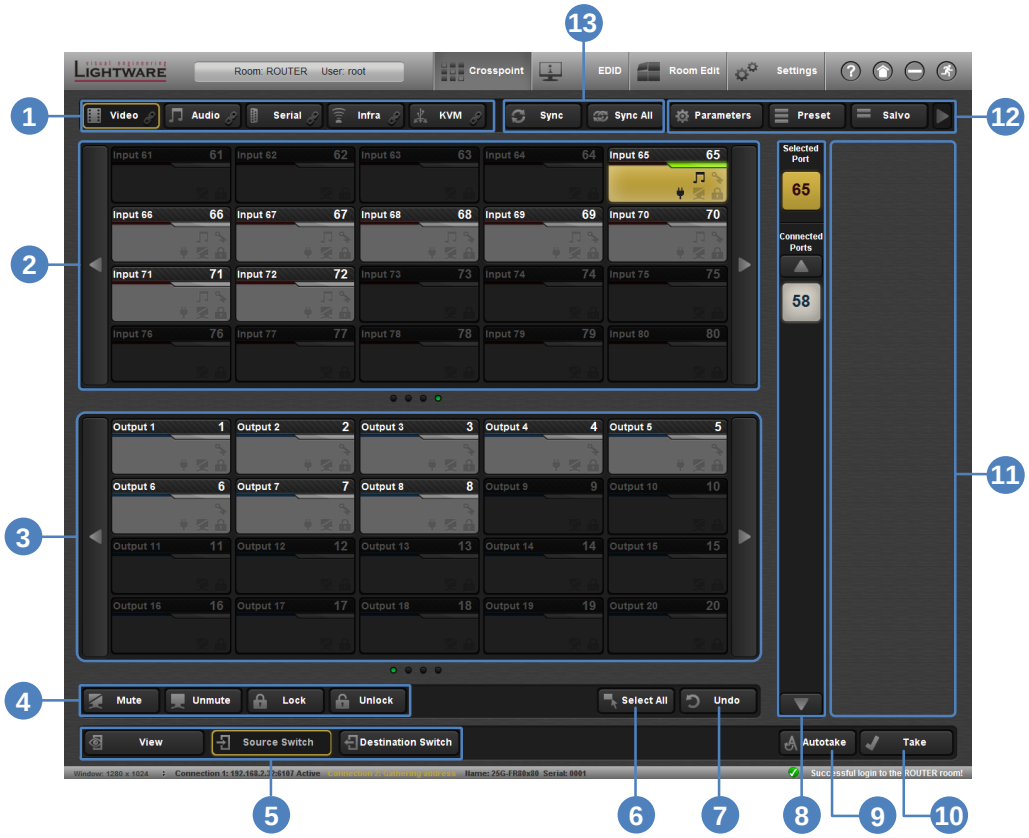


Figure 4-4. Crosspoint screen

1	Media layers	The available media layers are shown; the currently displayed one is highlighted by a yellow frame.
2	Input ports	Each button represents an input port. Pages can be turned by the small arrays on both sides. The actual page is signed by a green dot below, the other pages signed by black dots. (Depends on board numbers, frame and room size.)
3	Output ports	Each button represents an output port. Pages can be turned by the small arrays on both sides. The actual page is signed by a green dot below, the other pages signed by black dots. (Depends on boards numbers, frame and room size.)
4	Mute and Lock	Mute/Unmute and Lock/Unlock state can be set by the buttons.
5	View selector	View / Source switch / Destination switch modes are available. The active mode is highlighted by a yellow frame.
6	Select all	Selecting all input ports in ‘Destination switch’ mode; the button is changed to ‘Deselect’ when all ports are selected.
7	Undo	Withdrawing last activity in TAKE mode.
8	Selection panel	Displays the connection status of the selected port: listing the ports that are connected to it.
9	Autotake button	Toggles between TAKE/AUTOTAKE modes when ‘Source switch’ or ‘Destination switch’ view is selected.
10	Take button	Executes the modified crosspoint settings in TAKE mode.
11	Settings panel	Parameters, Preset or Salvo settings are displayed in this area.

12	Panel selector	The panel for Parameters, Preset or Salvo settings can be displayed. (Only one of them is visible at the same time.)
13	Sync buttons	Ports can be synchronized with other media layers by the buttons.

4.4.2. Buttons and symbols

Media layers

The available media layers can be controlled separately or linked together (synchronizing); the state of this function is visible on the button of the layer:

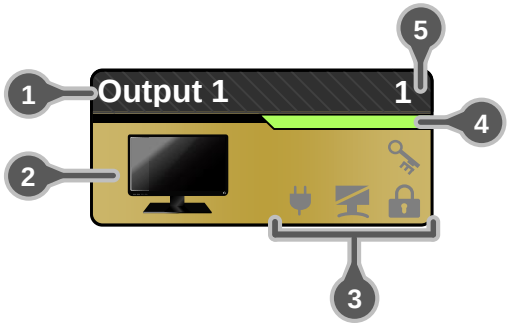
The layer is not linked to other layer(s) (chain is grey)

The layer is linked to other layer(s) (chain is highlighted)

Toggle the linking state by right clicking on the button (if touch screen is used, keep the button pressed). See more information about synchronizing in section 4.4.10 on page 30.

Port tiles

The colors of the port tiles and the displayed icons represent different states and information about selected port:



1	Port name
2	Background color and icon
3	State indicators
4	Signal present (green), not present (grey)
5	Port number

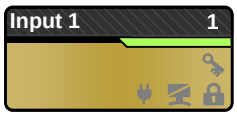
Info: The port name and the icon can be changed in the Room management menu.

Info: The displayed port number is not the same as the physical port number. This number is always counted from 1 and stepped by one in the given room.

Background color (port state)

The color of the I/O buttons determines the state as follows:

(Dark grey)
The port is not live



(Yellow)
The port is selected

(Light grey)
The port is live but not selected

(White)
The port is connected to the selected port

The port is not live (the button is dark grey) if

- The I/O board is powered down or does not work,
- No I/O board is installed at the given port, or
- The CPU does not control the board.

State indicators

The meaning of the icons are the followings:

Icon	Icon is not displayed	Icon is grey	Icon is black
	HDCP is not supported	HDCP is disabled	HDCP is enabled
	No information about connection status	No device is connected	Device is connected
	-	Port is unmuted	Port is muted
	-	Port is unlocked	Port is locked
	Embedded audio is not supported	Video signal does not contain embedded audio	Video signal contains embedded audio

4.4.3. Selection panel

The main function of the selection panel (Figure 4-4, item nr. 8.) is to show the number of selected (and connected) port(s) clearly since it is always visible, independently from the currently displayed page.

When a port is selected, its number is displayed on the top of the panel and the number of connected port(s) is/are displayed below it. If the space is not enough to display all port numbers, press the arrow buttons to scroll the list.

4.4.4. View mode

In 'View' mode the ports' state, crosspoint settings and other parameters can be checked without the risk that the crosspoint/lock/mute states are changed accidently. Switching, muting and locking functions are disabled.

Info: Salvo can be run in View mode that may modify the crosspoint, mute or locking states.

4.4.5. Take and Autotake

The router has two different switching modes: Take and Autotake.

Take mode is active

Autotake mode is active

Press the 'Autotake' button to toggle between Take and Autotake modes.

Take mode allows the user to connect or disconnect multiple outputs to an input at once. This mode is useful when time delay is not allowed between multiple switching. The commands are only realized when the Take button is pressed. The 'Take' button is disabled in 'Autotake' mode.

'Autotake' mode is useful when immediate actions must be done or fast switching is needed between sources on a particular destination. In this mode switching occurs immediately upon pressing one of the port selector buttons.

4.4.6. Source switch mode

In 'Source Switch' mode **one input port can be selected** that will be visible on the selection panel and one or more output ports can be switched to it.

Info: If a port is currently locked, neither its crosspoint, nor its muting state can be changed.

Changing connections

Step 1. Select an input port; its button will turn to yellow and its number will be displayed on the Selection panel. If one (or more) output port(s) is/are connected to it, the button(s) of the output port(s) will turn to white and will be displayed on the Selection panel as 'Connected Ports'.

Step 2. Select or deselect the desired output port(s).

- In **Take** mode: the button(s) of the output port(s) will start to blink.
- In **Autotake** mode: the (de)selected output ports will be (dis)connected to/from the input port immediately.

Step 3. Press the 'Take' button to execute changes (only in Take mode).

Info: All output ports can be selected by pressing the 'Select All' button. Pressing the button again (its label is changed to 'Deselect' in this case) deselects all output ports that were not connected previously.

4.4.7. Destination switch mode

In 'Destination Switch' mode **one output port can be selected** that will be visible on the selection panel and an input port can be switched to it.

Info: If a port is currently locked, neither its crosspoint, nor its muting state can be changed.

Changing connections

Step 1. Select an output port; its button will turn to yellow and its number will be displayed on the Selection panel. If an input port is connected to it, the button of the input port will turn to white and will be displayed on the Selection panel as 'Connected Ports'.

Step 2. Select or deselect the desired input port(s).

- In **Take** mode: the button of the input port will start to blink.
- In **Autotake** mode: the (de)selected input port will be (dis)connected to/from the output port immediately.

Step 3. Press the 'Take' button to execute changes (only in Take mode).

4.4.8. Muting input/output port(s)

When a port is muted, there is no signal transmission on it. The setting refers to the active and the synchronized layers.

Info: In 'Source switch' mode only the selected input ports can be (un)muted; in 'Destination switch' mode only the selected output ports can be (un)muted.

Step 1. Select the desired port(s).

Step 2. Press the 'Mute' or 'Unmute' button; the state is changed immediately.

Info: If the port is currently locked, its crosspoint and muting state cannot be changed.

4.4.9. Locking input/output ports

Locking feature ensures to avoid accidental/unwanted crosspoint changes. If the port is currently locked, its crosspoint and muting state cannot be changed. The setting refers to the active and the synchronized layers.

Info: In 'Source switch' mode only the selected input ports can be (un)locked; in 'Destination switch' mode only the selected output ports can be (un)locked.

Step 1. Select the desired port(s).

Step 2. Press the 'Lock' or 'Unlock' button; the state is changed immediately.

4.4.10. Synchronizing

Synchronizing is a new feature of 25G family. Basically the settings/changes of the crosspoint of the active media layer (that is displayed on the screen) is executed on all connected media layers (marked with green chain). The synchronizing can be switched on (chain is green) and off (chain is grey) by clicking on a media layer with the left button of the mouse (on the touch screen just keep the button pressed).

Basic rules of the synchronizing

- Synchronizing is not available in View mode.
- In Source Switch mode the input port means the selected port. In Destination Switch mode the output port means the selected port.
- Synchronizing is executed only if the active layer is also selected (chain is green).
- Synchronizing has an effect on the crosspoint settings (connected ports to a selected port), and on mute/lock state of a port.
- Changes are executed only on those ports, which are available for the user.
- Undo button can be used to withdraw the operations. After pressing the Undo button, the crosspoint settings will be withdrawn on the active- and all affected layers.
- Synchronizing is not executed on those layers where the signal directions on the port are different. E.g. a video input port and return audio port cannot be synchronized.

There are three working modes of synchronizing:

Simple mode

In this case any crosspoint setting that is done on the active layer, automatically executed on connected layers, too. Previous settings of the active layer are not synchronized to connected layers.

Step 1. Mark the desired media layers by the right button of the mouse (if touch screen is used, keep the layer button pressed).

Step 2. Select a layer that will be the active one (its settings will be synchronized to the connected layer(s)).

Step 3. Set the desired crosspoint layout – they will be executed automatically on connected layer(s):

- In **Autotake** mode the changes are made immediately on all layers,
- In **Take** mode the changes are made after pressing the Take button.

Sync mode

This mode can be used to synchronize the crosspoint settings of a certain port.

Step 1. Mark the desired media layers by the right button of the mouse (if touch screen is used, keep the layer button pressed).

Step 2. Select a layer that will be the active one (its settings will be synchronized to the connected layers).

Step 3. Press the Sync button and the followings will be executed on connected layer(s):

- Crosspoint change: connected ports of the selected port
- Mute/lock state of selected port will be the same on all layers.

Sync All mode

The mode is similar then the Sync mode, but in this case all the ports are synchronized, which are visible on the Source/Destination panel.

Step 1. Mark the desired media layers by the right button of the mouse (if touch screen is used, keep the layer button pressed).

Step 2. Select a layer that will be the active one (its settings will be synchronized to the connected layers).

Step 3. Press the Sync All button and the followings will be executed on connected layer(s):

- Crosspoint change: connected ports of the selected port
- Mute/lock state of selected port will be the same on all layers.

4.4.11. Parameters panel

Press the 'Parameters' button on the Settings panel to display more information about the selected port. (Press the button again to hide the panel.) Displayed settings and information are different port by port.

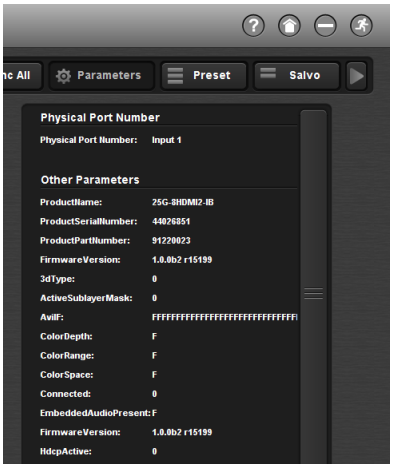


Figure 4-5. Port parameters panel

4.5. EDID menu

4.5.1. About EDID memory

EDID memory is non-volatile and consists of four blocks, each for different purposes:

EDID type	Description	Number of EDIDs
Factory EDID	Factory preset EDID, cannot be changed	97
User EDID	User programmable memory	100
Dynamic EDID	Last Attached Monitors' EDID on a specific output port	160*
Emulated EDID	EDID currently emulated on a specific input port	160*

* Depends on the capabilities of the matrix and the installed boards.

The list of factory EDIDs are listed in section 5.5.11 on page 64.

Info: Both 128 Byte EDID and 256 Byte extended EDID structures are supported.

Info: The attached monitor's EDID is stored automatically, until a new monitor is attached to that particular output. In case of powering the unit off, the last attached monitor's EDID remains in non-volatile memory even if the monitor is unconnected.

4.5.2. EDID menu layout

Advanced EDID Management can be accessed by selecting the EDID menu.

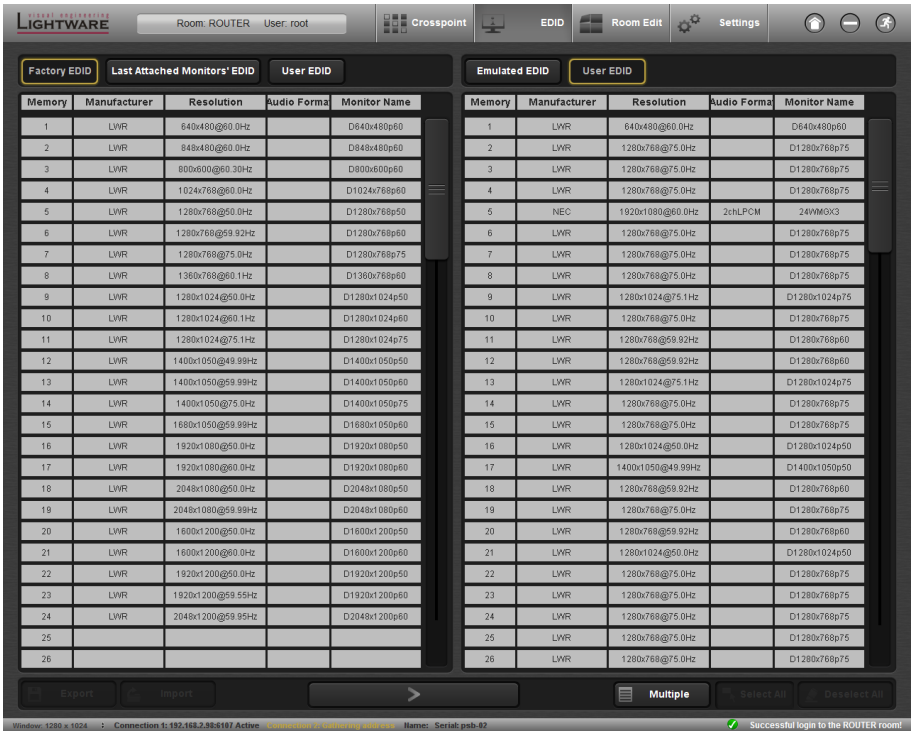


Figure 4-6. EDID management

The view is divided in two segments: the left panel contains the EDIDs that can be used as a source, the right panel contains the target places where the EDIDs can be emulated. The same EDID can be emulated or copied to one or more ports by using the 'Multiple' button.

OFF: one EDID can be selected in the list on the right panel

ON: more EDIDs can be selected in the list on the right panel

EDID emulation/copy can be executed by the 'Transfer' button:

4.5.3. Changing the emulated EDID at one input

Step 1. Select the desired EDID list from one of the three sources by pressing its button above the left panel.

Step 2. Select an EDID from the list on the left panel that has to be emulated; the EDID will be highlighted by yellow cursor.

Step 3. Press the 'Emulated EDID' button above the right panel.

Step 4. Select the desired input number where the EDID has to be emulated; the EDID will be highlighted with yellow cursor.

Step 5. Press the 'Transfer' button to emulate the EDID.

4.5.4. Changing the emulated EDID at more inputs

Step 1. Press the 'Emulated EDID' button above the right panel.

Step 2. Select the desired EDID list from one of the three sources by pressing its button. The EDIDs will be listed on the left panel.

Step 3. Press the 'Multiple' button if it is in OFF state. Select the target memories in the list on the right panel. You can select

- one by one with single clicks,
- more memories by keeping the mouse button pressed or
- all the memories by pressing the 'Select All' button.

Step 4. Press the 'Transfer' button to emulate the EDIDs.

4.5.5. Learning EDID

Info: The process is the same like changing the emulated EDID, the only difference is the target on the right panel: press the 'User EDID button'. Thus one or more EDIDs can be copied into user memory.

4.5.6. Exporting EDIDs

The control software is able to download an EDID listed on the left panel and to save it as an EDID file (*.dat - recommended, *.bin or *.edid file extension can be also selected).

Step 1. Select the desired EDID from the list on the left panel.

Step 2. Press the 'Export' button. A window will pop up; select the target folder, the desired format and press the 'Save' button.

4.5.7. Importing EDIDs

Previously saved EDIDs can be imported and stored in the User memory as follows:

Step 1. Press the 'User EDID' button above the left panel.

Step 2. Select the desired EDID memory place where the EDID will be stored.

Step 3. Press the 'Import' button below the left panel. In the appearing window browse the EDID file, select it and press the 'Open' button.

Step 4. If the EDID is valid, it will be stored in the desired User memory.

4.5.8. Deleting EDIDs

This feature is under development.

4.6. Room Edit

The room management is one of the most important concept of the 25G. Rooms can be imagined as virtual routers which can perform every operation what is expected from a router (like switching, saving and running salvos, inject commands, set up various parameters, etc.). The I/O ports of these virtual routers may be (but not necessarily) assigned to the real physical ports, however the correspondence can be removed or altered any time by the system administrator.

A room may have one or more media layers included. It is possible to define rooms with only one layer (e.g. only video layer), but it is also possible to add every layer. Layers in the room can be handled (switch, mute, lock, etc.) independently, but they can be handled also together. For clarity reasons, the physical ports must be the same on every layer in a room, however it is possible to skip ports on specific layers and leave it unassigned.

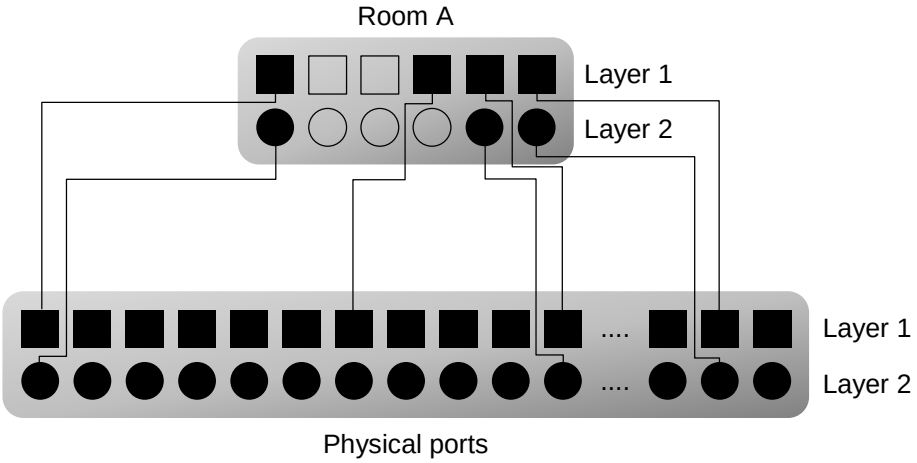


Figure 4-7. Room management

If a user makes any change in a room, then the corresponding (assigned) physical ports and the crosspoint will follow the change. The main advantages of the rooms that their internal port numbering is independent from the physical ports – in an installed system the external controller doesn't need to know where are the real sources and sinks connected to, and by working inside the room it is impossible to affect other, external ports. That way multiple controllers can be operated on the same router simultaneous safely, while the system administrator can change the cabling any time without reprogramming them. It is also even possible to create new unassigned, purely virtual rooms where the controllers can be programmed and debugged without any external connected device.

It is important that rooms can be overlapped and therefore they may interact each other. If a port is present in two or more rooms, then every port related operation (e.g. connect to other ports, change the settings) will affect the port assigned to the same physical port in every other rooms.

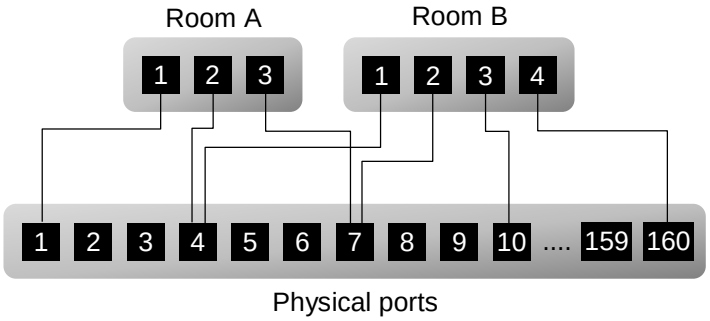
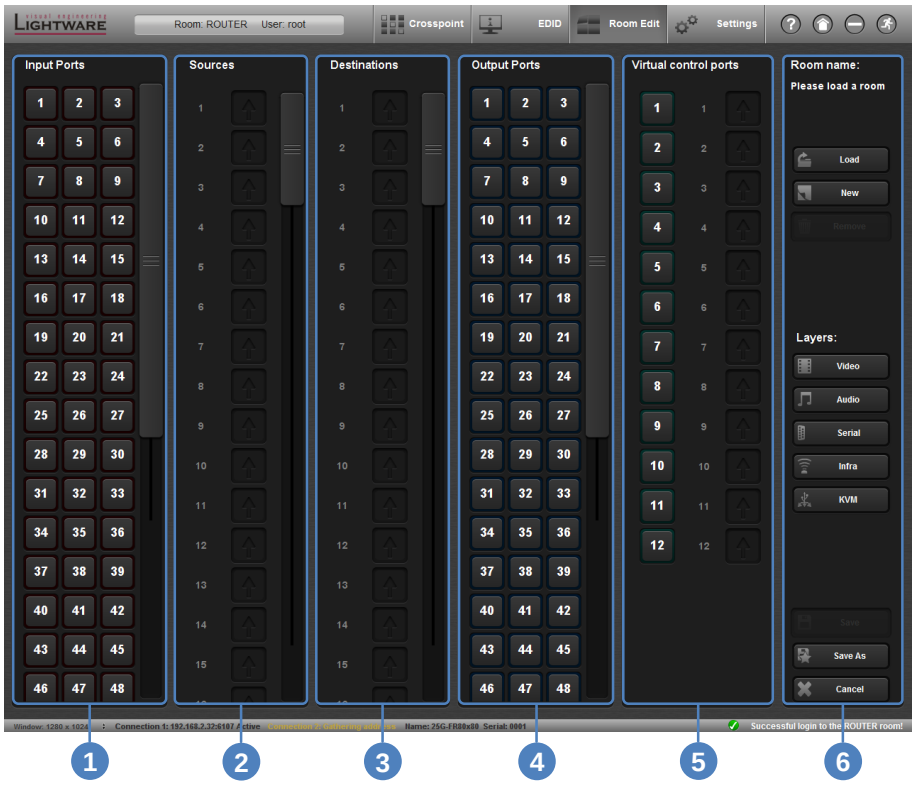


Figure 4-8. Rooms – overlapping

4.6.1. Room Edit window layout



1	Input ports	All the input ports which are available in the matrix. These ports can be selected to the rooms as input sources.
2	Sources	These are the ports which will be available as input ports in the defined room.
3	Destinations	These are the ports which will be available as output ports in the defined room.
4	Output ports	All the output ports which are available in the matrix. These ports can be selected to the rooms as output destinations.
5	Virtual control ports	Ports which can be used for e.g. command injection functions.
6	Control panel	Control buttons for room editing operations.

4.6.2. Creating a room

Step 1. Navigate to the Room Edit menu.

Step 2. Press the New button on the right side. In the opening window type the desired room name and press OK. (If the name already exists, an error message appears.)

Step 3. Select the desired media layers on the Control panel.

Info: The selected port will appear on all marked layers. E.g. Audio and Video layers are marked, input port nr. 1 is added; Input 1 will be available on the Video and Audio layers.

Step 4. Click on the desired input ports, thus they will be mounted to the room as Sources. (Drag and drop the Sources to the Input ports to remove.)

Step 5. Click on the desired output ports, thus they will be mounted to the room as Destinations. (Drag and drop the Destinations to the Output ports to remove.)

Step 6. Click on the desired virtual control ports and add them to the room.

Step 7. Press the Save button to store the settings.

Info: Virtual control ports can be used when serial devices are desired to control by sending commands via LAN. In this case the communication is performed through the virtual port.

4.6.3. Editing a room

Step 1. Navigate to the Room Edit menu.

Step 2. Click on the Load button. Select the desired room in the opening window and click on OK. The room layout will be loaded to the panels.

Step 3. Select the media layers, then drag and drop the desired ports between the panels.

Step 4. Store the settings by pressing

- The **Save** button: the room will be saved to the original name, or
- The **Save as** button; type a new name in the opening window and press OK.

4.6.4. Deleting a room

Step 1. Navigate to the Room Edit menu.

Step 2. Click on the Load button. In the opening window select the desired room and click on OK. The room layout will be loaded to the panels.

Step 3. Click on 'Remove' button on the control panel.

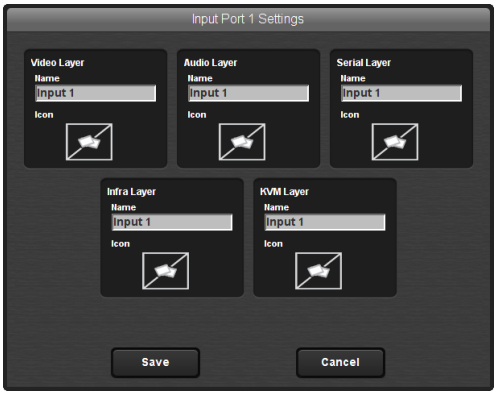
Step 4. Press the 'Yes' button in the appearing window if you are sure about deleting the room. The room is going to be deleted.

4.6.5. Change the name and icon of a room

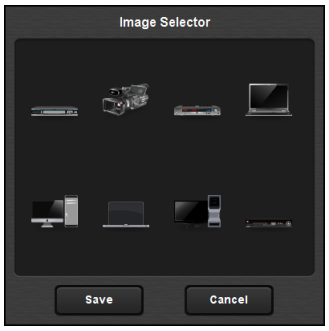
Names and icons of the ports can be customized in the room editor. The names and icons cannot be different in rooms as they are assigned to the input and output ports.

Step 1. Navigate to the Room Edit menu.

Step 2. Click on the desired port by the left button of the mouse (when touch screen is used, keep the button pressed). A new window will pop up.



Step 3. Different icon and name can be defined to the layers within the port. Name the layers and select the icon by clicking on the pictogram. (Click again to deselect.)



Step 4. Press the Save button in the window(s).

4.7. Settings

4.7.1. Communication Settings

The tab shows the CPU's and the Single Board Computer's LAN settings. If anything has been modified on a panel, press the 'Save' button on it to store the settings.

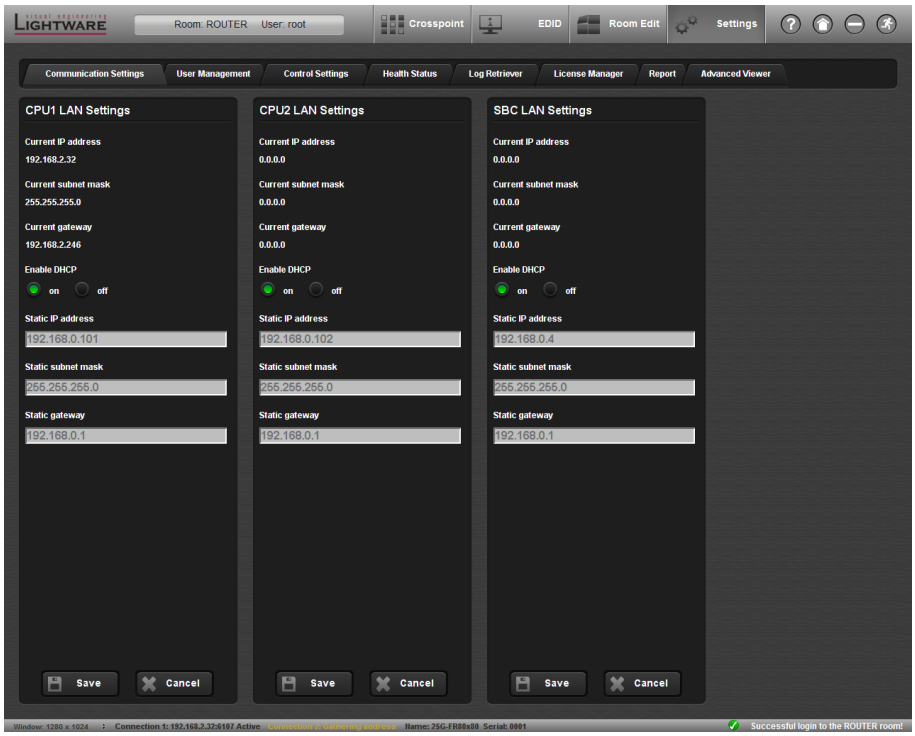
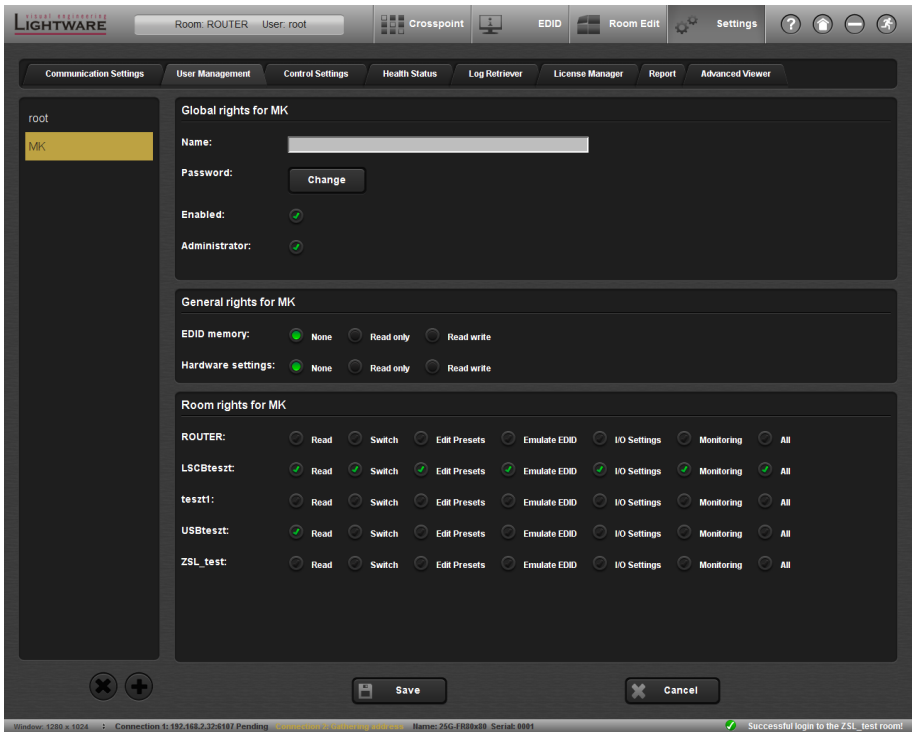


Figure 4-9. Communication Settings tab

4.7.2. User management

A new feature of 25G family is the User management. Different users can be created with different rights; navigate to this tab in the Settings menu.



Defined users are listed on the left panel, settings are displayed on the right panels.

Info: User management is available only for the users with Administrator rights.

Global rights

User's name and password can be changed on this panel.

Enabled: The user may login if this option is ticked.

Administrator: The user may modify the users if this option is ticked.

General rights

EDID memory: Access to the 'EDID' menu and settings.

Hardware settings: Access to the 'Settings' menu.

Room rights

Defined rooms are listed which can be set individually to the users.

Adding a new user

Step 1. Click on the '+' icon below the left users' list.

Step 2. A window will pop up; type the desired name and click OK.

Step 3. Set the desired rights to the user (default settings are rather limited).

Deleting an existing user

Step 1. Select the desired user from the users' list.

Step 2. Press the 'x' button below the list.

Step 3. A confirmation window will pop up; press the 'Yes' button.

About root user

Root user is defined as factory default setting. It cannot be deleted, and root's rights cannot be modified or seen by other users. Root user is Administrator and has full read/write rights to the settings and rooms.

4.7.3. Control Settings

External controllers (e.g. touch panel) can be connected to the matrix by RS-232 or Ethernet port. The settings are available in this submenu.

Add a new controller

Delete the selected controller

Each defined controller can be enabled or disabled according to the requirements.

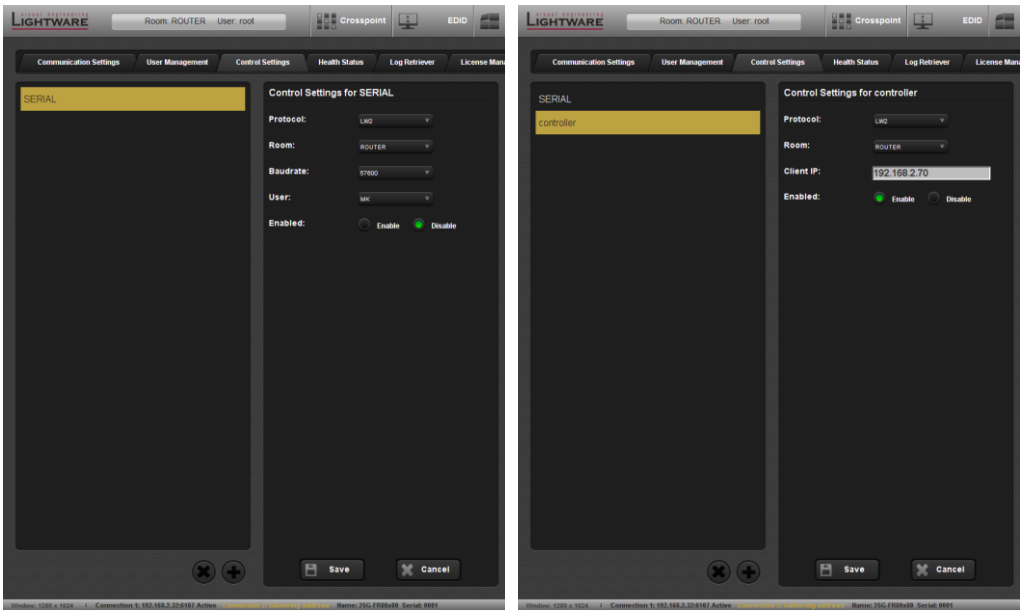


Figure 4-10. Serial and Ethernet Controller settings

Serial port settings

If the controller is connected via RS232 the protocol has to be set:

- If LW2 or P#2 is selected, a room also has to be assigned,
- If LW3 is selected, a user also has to be assigned.

Please also set the Baud rate that is in line with the controller.

Ethernet port settings

If the controller is connected via Ethernet, the followings have to be set:

- The desired protocol (LW2 or P#2),
- The desired room (the controller will work only in the selected room), and
- The IP address of the controller (Client IP).

4.7.4. Health Status

Device information, such as installed boards and hardware health are displayed in this submenu. Press the 'Details' button or click on the picture to see these information about the unit.

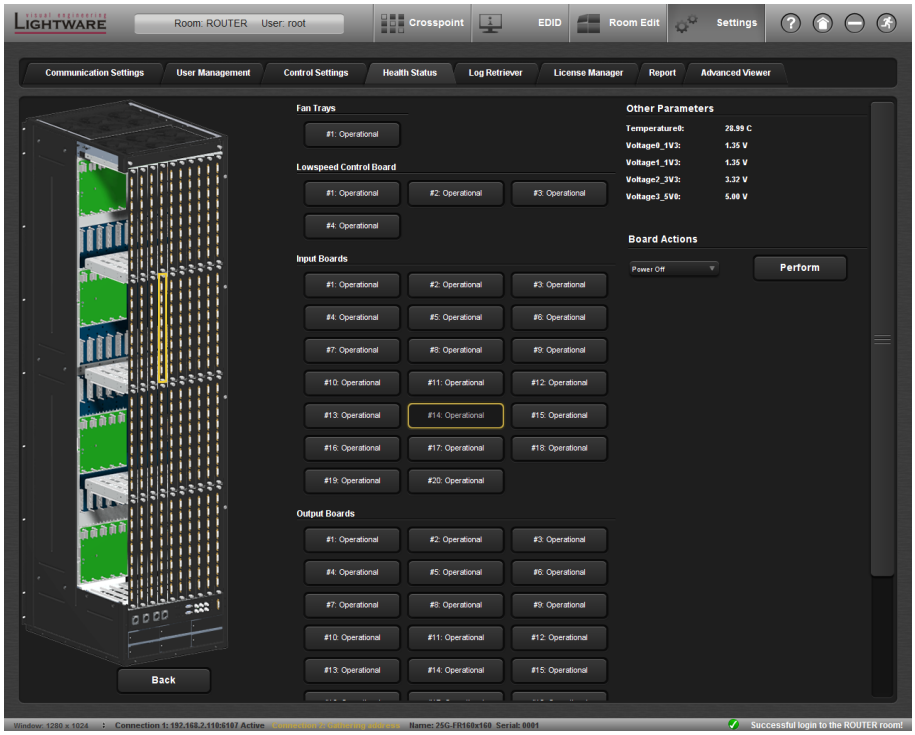


Figure 4-11. Health Status submenu

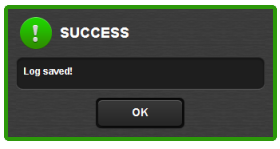
The tab displays information about:

- The board's status (operational/empty),
- Voltage levels,
- Fans' speed and
- Temperature values.

Boards can be powered down or reset – if the feature is supported by the board.

4.7.5. Log Retriever

All the events logged by the matrix can be downloaded into a log file in this submenu. You can limit the number of log rows, and date interval can also be set. Press the 'Save..' button to start saving the log file to your PC.



Info: Please note that the saving process may take some minutes since all events are collected and generated into a file.

4.7.6. License Manager

The 'License Manager' is a surface where new additional features can be applied. If a new board or function is purchased from Lightware that had not been in the original device, a license may be applied. License codes are generated by Lightware and valid only for the given device. If such a code is necessary to insert, Lightware will inform you about the procedure.

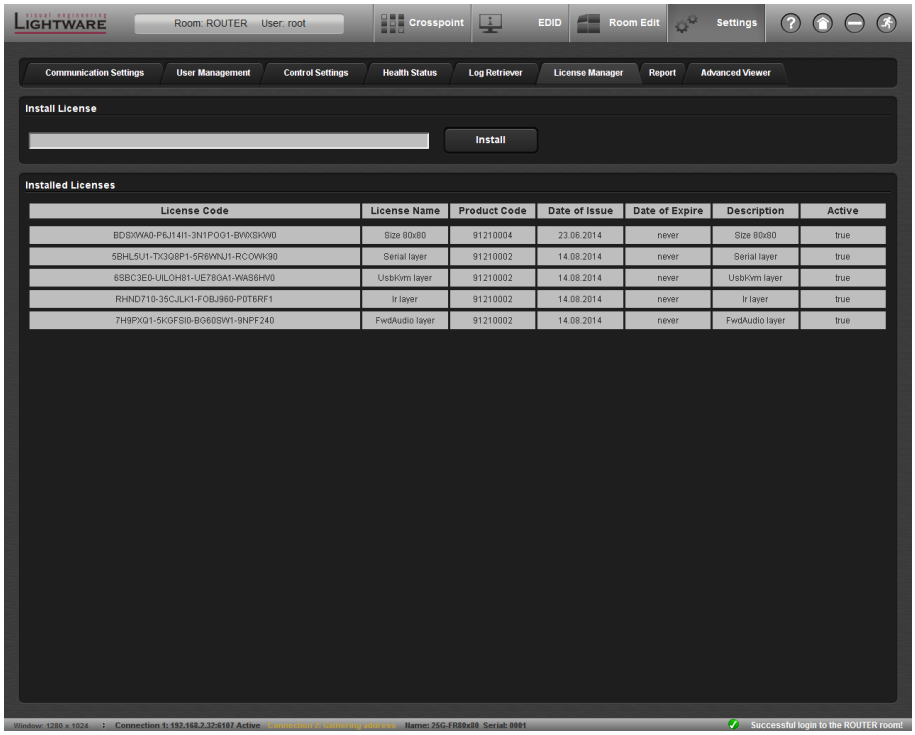


Figure 4-12. License Manager

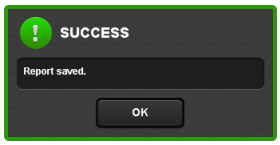
4.7.7. Report tab

The tab is the surface for the well-known (from Lightware Matrix or Device Controller Software) report generator feature. In this case a ZIP file is generated which contains all the collected log and reports files.

Step 1. Press the 'Generate..' button; the 'Save report..' window will pop up.

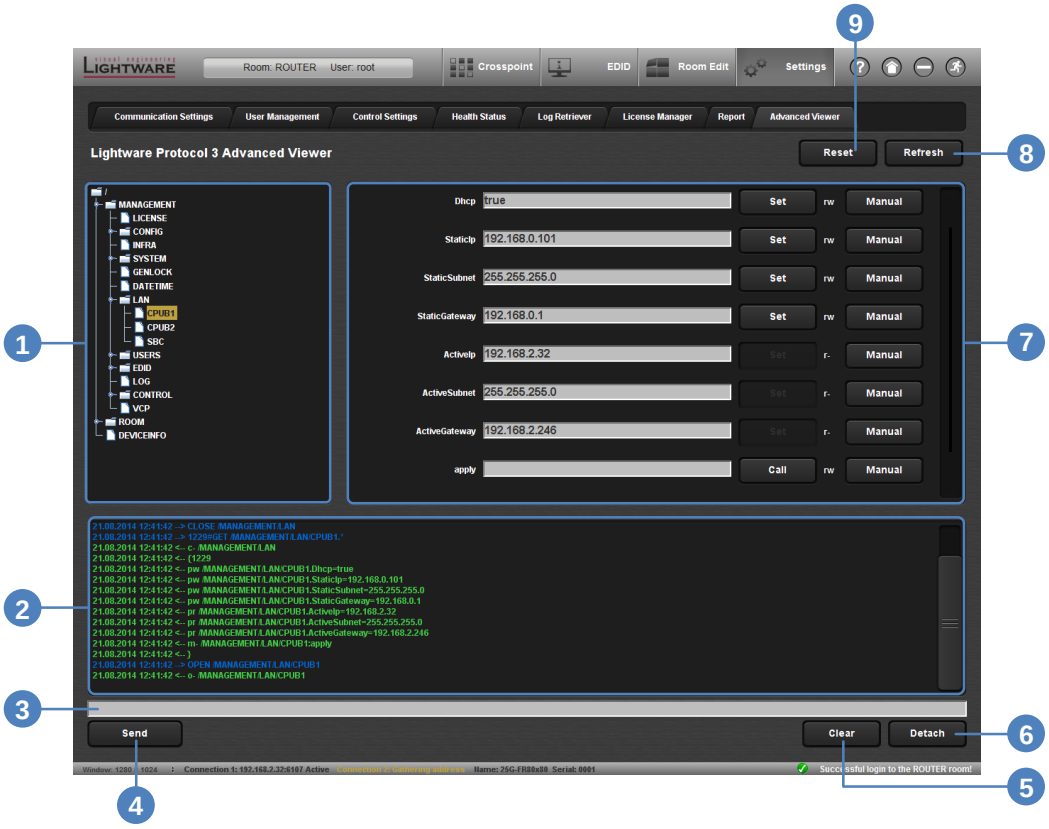
Step 2. Select the desired destination, type the file name and press the 'Save' button.

Step 3. The matrix will collect all the necessary information. When that is done, a confirmation window will pop up.



4.7.8. Advanced Viewer

The tab is the advanced surface for displaying the structure and the elements of node tree – defined in Lightware Protocol 3. Terminal commands and specific parameters (that are not available on the user interface) can be called and set. Detailed information about LW3 protocol, nodes, methods and parameters can be found in chapter 5 on page 43.



4-13. Figure Advanced viewer

1	**Protocol tree**	The whole LW3 protocol tree is available via this panel. Select an item to see its content.
2	**Log window**	Commands and responses are listed in this window. The timecode in every row shows the exact time when the command was sent or the response received.
3	**Command line**	Command texts can be typed directly by this line.
4	**Send button**	Press the button or hit the Enter to execute the command typed in the Command line.
5	**Clear button**	Clears the content of the log window.
6	**Detach log window**	The log window can be displayed on a separate panel thus keep opened beside the main window.
7	**Parameters' list**	Shows the correspondent parameters and methods which belong to the selected node in the protocol tree.
8	**Refresh button**	Reloads the content of the currently opened item.
9	**Reset button**	Reloads the content of the whole tree and set the focus to the default item (root node).

5. Programmer's reference

5.1. LW3 protocol – Overview

Lightware 3 (LW3) protocol is currently used by the 25G product line, the MODEX extender family and will be the preferred protocol in the new developments.

The LW3 is an ASCII based protocol and all commands are terminated with a carriage return (Cr, '\r') and line feed (Lf, '\n') pair. It is organized as a tree structure that provides outstanding flexibility for implementing a human readable, but still easy to programmatically parse protocol, which is suitable for different products with different feature list.

The concept

In order to implement a flexible, easy-to-use protocol that is straightforward to adapt to new devices and provides outstanding scalability and sustainability, we decided to organize all settings, parameters and properties of the device to a tree structure with 'nodes', 'properties' and 'methods'.

5.1.1. Elements of tree structure

Info: All names and values are case sensitive. The space character is replaced by '●' charcter in the elements and commands descriptions.

Node

- The basic building block of the tree structure is the 'node'.
- A node can have multiple child nodes, but only one parent.
- The tree has only one root the 'root node'.
- The leaves of the tree are also nodes, which do not have child nodes.
- The nodes are separated by a slash ('/') character.
- All the slashes are 'right slashes', no backslash is used.
- The identifier of the root node it a slash ('/')
- Nodes' name can contain the elements of the English alphabet and numbers.
- Recommended convention for case sensitivity:
  - Fix nodes (that cannot be altered) are capitalized.
  - User created nodes can contain both lowercase and capital letters, no restrictions.
- The path of a node has to contain all parent nodes from the root node.

Format (the root node): nX●/

Path: nX●/[nodeName]/[nodeName]/[nodeName]

Legend: **'**n': node

'X' can be:

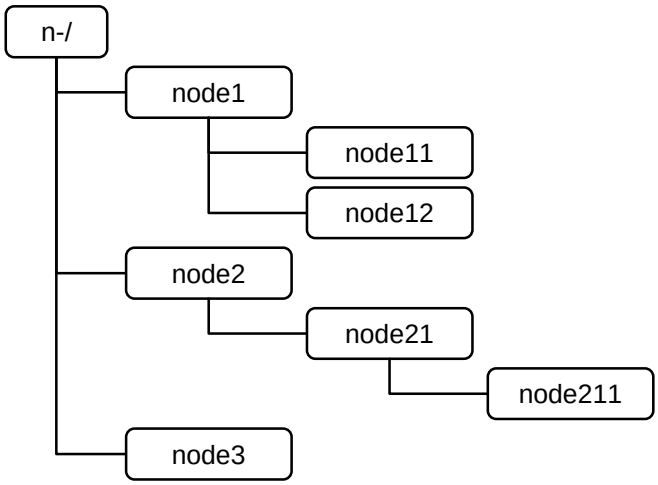
'-': default for a node,

'm': this is a manual for the node (see section 5.2.10 on page 53),

'E': this is an error message for the node (see section 5.1.3 on page 45).

Info: All parent nodes must be listed in the path of a node.

The example below presents the depth tree traversal of Figure 5-1:



Path of the nodes:

n-/node1
n-/node1/node11
n-/node1/node12
n-/node2
n-/node2/node21
n-/node2/node21/node211
n-/node3

Figure 5-1. Tree structure of nodes

Property

The 'property' in the LW3 protocol is basically a leaf, which has a well-defined value.

- A property has a value.
- A property cannot have child nodes or child properties. It is always a leaf.
- A node can have any number of properties (may not have any).
- A property is referenced with a dot ('.') after the node name.
- The properties' name can contain the elements of the English alphabet, numbers and underscore ('_') character.
- By convention, properties are beginning with capital letter, all other characters are lowercase ones. In case of compound words, all words are beginning with a capital letter (CamelCase).
- The value of the property can contain any readable ASCII character (the control characters have to be escaped, see section 5.1.2 on page 45).
- A property can be read-only or read/write.

Format: pX●/[nodeName].[propertyName]=[propertyValue]

Legend: p: property

X can be:

'r': if the property is read-only.

'w': if the property is readable, writable.

'm': manual for the property (see section 5.2.10 on page 53).

'E': error message for the property (see section 5.1.3 on page 45).

Example:

The following two ones are read-only properties:

```
<-- pr●/node1/node12.ReadOnlyProperty=value1
<-- pr●/.DeviceName=25G Hybrid Device
```

The following two ones are read-write properties:

```
<-- pw●/node1/node12.ReadWriteProperty=value2
<-- pw●/.DeviceNickName=John
```

Method

The 'method' in the LW3 protocol is also a leaf. It cannot have a value, such as the properties, but it can be invoked with a parameter with the help of a special 'CALL' command (see section 5.2.5 on page 49).

- A method cannot have child nodes or child methods. It is always a leaf.
- A node can have any number of methods (may not have any).
- A method is referenced with a colon (':') after the node.
- The methods' name can contain the elements of the English alphabet, numbers and underscore ('_') character.
- By convention, methods are beginning with lowercase letter. In case of compound words, the very first letter is lowercase, and the first letter of each other words are capitalized (lowerCamelCase).
- The parameter of the method can contain any readable ASCII character (the control characters have to be escaped, see section 5.1.2 on page 45).
- The method always has a return 'state' if the method could be executed. The state could be either 'OK' or 'FAILED'.
- The method not necessarily has a return 'value'. If it does, it can contain additional information, which is always specific for the current case (the return value can specify why the execution failed). Find details in the section 5.2.5 on page 49.
- When the method cannot be executed (e.g. the parameter list is illegal), there is an error message (see section 5.1.3 on page 45).

Format: mX●/[nodeName]:[methodName]=[returnValue]

Legend: m: method

X can be:

'O': when the execution of the method was successful,

'F': when the execution of the method failed,

'm': manual for the method (see section 5.2.10 on page 53),

'E': error message for the method (see section 5.1.3 on page 45).

Example:

```
<-- mO●/node1/node12:method1
<-- mO●/MANAGEMENT/USER/getEmail=john@john.com
<-- mF●/MANAGEMENT/USER/getEmail=The user John does not exist
```

5.1.2. Escaping

Property values and method parameters can contain characters that are used as control characters in the protocol. They must be escaped. The escape character is the backslash ('\') and escaping means injecting a backslash before the character that should be escaped (like in C language).

Control characters are the followings: \ { } # % ( ) \r \n \t

Example:

The original text: John●(Doe).●#3:●5%2=1●node1\node11

The escaped text: John●\(Doe\).●\#3:●5\%2=1●node1\\node11

5.1.3. Error messages

There are several error messages defined in the LW3 protocol. All of them have a unique error number which can be used programmatically, and an informative error message.

Format: XE●[primitive]●%EYYYY:●[Error message]

Legend: X can be:

‘-’: syntax error. Cannot parse the command at all.

‘n’: node error.

‘p’: property error.

‘m’: method error.

YYY: error code, which can be one of the followings:

E001	Syntax error	E010	Illegal parameter count
E002	Node not found	E011	Item already exist
E003	EOL expected	E012	Item does not exist
E004	Writing read-only property	E013	Illegal operation
E005	Invalid value	E014	Internal error
E006	Property does not exist	E015	Access denied
E007	Syntax error	E016	Write access denied
E008	Illegal method	E017	Reserved
E009	Method does not exist	E018	Waiting timeout

5.1.4. Prefix summary

The following prefixes are defined in the LW3 protocol:

‘n-’: a node,

‘nE’: an error for a node,

‘nm’: a manual for a node,

‘pr’: a read-only property,

‘pw’: read-write property,

‘pE’: an error for the property,

‘pm’: a manual for the property,

‘m-’: a method,

‘mO’: a response after a success method execution,

‘mF’: a response after a failed method execution,

‘mE’: an error for a method,

‘mm’: a manual for a method.

5.2. Commands

Getter

The ‘GET’ command can be used to get the child nodes, properties and methods of a specific node. It also can be used to get the value of a property.

The response format

The first two characters of a response are unambiguously identifies the type of the element that the response line concerns. The first character is the type of the element (node/property/method), the second is for miscellaneous information (e.g. read/write rights).

The defined prefixes are:

'n-': node

'pr': property - only readable

'pw': property - writable, readable

'm-': method executable

After the prefix the response contains the full path of the node, property or method after a space character.

5.2.1. Get all children of a node

Get all of the child nodes of a parent node, with one GET command.

Command format: GET●[nodePath]

Response format: n-●[nodePath]

Example:

```
--> GET●/ROOM/roomName/XP
<-- n-●/ROOM/roomName/XP/VIDEO
<-- n-●/ROOM/roomName/XP/AUDIO
<-- n-●/ROOM/roomName/XP/RS232
<-- n-●/ROOM/roomName/XP/INFRA
<-- n-●/ROOM/roomName/XP/USBKVM
```

5.2.2. Get all properties and methods of a node

Get all properties and methods of a specific node, with one GET command, using an asterisk wildchar.

Command format: GET●[nodePath].*

Response format: (for properties)

pX●[nodePath].[propertyName]=[parameter]

Legend: X can be:

'r': read-only

'w': read-write

Response format: (for methods)

m-●[nodePath]:[methodName]

Example:

```
--> GET●/ROOM/roomName/XP.*
<-- pw●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I5
<-- m-●/ROOM/roomName/XP:switch
<-- m-●/ROOM/roomName/XP:switchMulti
<-- m-●/ROOM/roomName/XP:mute
```

5.2.3. Get all child nodes, properties and methods of a node

Get all child nodes, properties and methods of a node with one command, without using a wildchar.

Command format: GETALL●[nodePath]

Response format: (for nodes)

n-●[nodePath]

Response format: (for properties)

pX●[nodePath].[propertyName]=[parameter]

Legend: X can be:

'r': read-only

'w': read-write

Response format: (for methods)

m-●[nodePath]:[methodName]

Example:

```
--> GETALL●/ROOM/roomName/XP
<-- n-●/ROOM/roomName/XP/VIDEO
<-- n-●/ROOM/roomName/XP/AUDIO
<-- n-●/ROOM/roomName/XP/RS232
<-- pw●/ROOM/roomName/XP.ST=I1,I2,I5
<-- m-●/ROOM/roomName/XP:switch
<-- m-●/ROOM/roomName/XP:switchMulti
<-- m-●/ROOM/roomName/XP:mute
```

5.2.4. Set command

The setter command can be used to modify the value of a property.

Command format: SET●[nodePath].[propertyName]=[newPropertyValue]

Response format:

The response for setting a property to a new value is the same as the response for the 'GET' command. The value in the response is the new value if the execution of the 'SET' command was successful, otherwise the unmodified 'old value' with an error message.

pw●[nodePath].[propertyName]=[newPropertyValue]

Example:

```
--> SET●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I5
<-- pw●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I5
```

Error response format:

If there were errors during setting a property, an error message follows the unmodified property value. Find the error numbers in the section 5.1.3 on page 45.

pE●[nodePath].[propertyName]=[umodifiedValue]●%EXXX:Error message

Legend:

XXX: error number (see section 5.1.3 on page 45).

Examples:

```
--> SET●/ROOM/roomName/XP.ReadOnlyProperty=true
<-- pE●/ROOM/roomName/XP.ReadOnlyProperty=false●%E004:Writing●read-only●property
--> SET●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I8
<-- pE●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I5●%E002:
Access●denied
--> SET●/ROOM/roomName/XP.DestinationConnectionStatus=true
<-- pE●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I5●%E003:
Invalid●value
--> SET●/ROOOM/roomName/XP.DestinationConnectionStatus=true
<-- pE●/ROOOOM/roomName/XP.DestinationConnectionStatus●%E002:
Node●not●found
```

5.2.5. Invocation

A method can be invoked with the help of the 'CALL' command.

Command format: CALL●[nodePath]:[methodName]([parameter])

Response format:

The response for a method execution is a state and a value. The state is mandatory and always defined, if the method could be executed. It can be either a success or a failure. The value is optional and it can contain additional information, such as the reason why the state is a failure or a specific value when the state is success, that the client can process. It is also possible to get an error message, when the method could not be executed – e.g. the parameter was illegal - and hence not even the state of the execution could be specified.

mX●[nodePath]:[methodName]=Y

Legend: X can be:

'O': if the execution is successful.

'F': if the execution is failed, but the method could be executed.

'E': if the method could not be executed: e.g. illegal parameter count.

Y can be:

- the return value of the method if any.
- it is valid that a method does not have any return value. In this case the equal sign ('=') can be omitted.

Example:

```
--> CALL●/ROOM/roomName/XP:switch(I1,O2)
<-- mO●/ROOM/roomName/XP:switch
--> CALL●/ROOM/roomName/XP:switch(I1,O3)
<-- mF●/ROOM/roomName/XP:switch=O3●is●locked
```

Error response format:

If there were errors during the execution, an error message is received, which follows the method name. Find the error numbers in the section 5.1.3 on page 45.

mE●[nodePath]:[methodName]●%EXXX:Error message

Example:

```
--> CALL●/ROOM/roomName/XP:switch(false)
<-- mE●/ROOM/roomName/XP:switch●%E009:Illegal●parameter●count
--> CALL●/ROOM/roomName/XP:switchhh(I1,I2)
<-- mE●/ROOM/roomName/XP:switchhh●%E008:Method●not●exists
```

5.2.6. Subscription

A user can subscribe to any node. Subscribe to a node means that the user will get a notification if any of the properties of the node is changed. These notifications are asynchronous messages - such as the ones described above - and hence they are useful to keep the client application up-to-date, without receiving any unwanted information. When the user does not want to be informed about the changes anymore, he can simply unsubscribe from the node.

Info: The subscriptions are handled separately for connections and not to users. Hence, if the connection is terminated all registered subscriptions are deleted. After every connection the subscribe command has to be sent in order to get the notifications of the changes.

Subscribe to a node

Command format: OPEN●[nodePath]

Response format: o-●[nodePath]

Example:

```
--> OPEN●/ROOM/roomName/XP
<-- o-●/ROOM/roomName/XP
```

Subscribe to multiple nodes

In order to subscribe to multiple nodes, the asterisk wildchar can be used.

Command format: OPEN●[nodePath]/*

Response format: o-●[nodePath]/*

Example:

```
--> OPEN●/MANAGEMENT/LOG/*
<-- o-●/MANAGEMENT/LOG/*
```

Get the active subscriptions

Issuing an ‘OPEN’ command without any parameters returns the active subscriptions for the current connection.

Command format: OPEN

Response format: o-●[nodePath]

Example:

```
--> OPEN
<-- o-●/ROOM/roomName/XP
<-- o-●/MANAGEMENT/LOG/WARNING
<-- o-●/MANAGEMENT/LOG/NOTICE
```

Unsubscribe from a node

Command format: CLOSE●[nodePath]

Response format: c-●[nodePath]

Example:

```
--> CLOSE●/ROOM/roomName/XP
<-- c-●/ROOM/roomName/XP
```

Unsubscribe from multiple nodes

In order to subscribe to multiple nodes, the asterisk wildchar can be used.

Command format:

CLOSE●[nodePath]/*

Response format

c-●[nodePath]/*

Example:

```
--> CLOSE●/MANAGEMENT/LOG/*
<-- c-●/MANAGEMENT/LOG/*
```

5.2.7. Notifications about the changes of the node structure

When the child node structure to which the user is subscribed is changed, an asynchronous notification is generated. In this case the notification is similar to a node response. The following changes in the structure trigger notification:

- a child node was created,
- a child node was deleted,
- a child node was renamed.

New node created

Format: NEW●[nodePath]

Example:

```
--> NEW●/MANAGEMENT/USER/JohnDoe
```

Node was deleted

Format: DEL●[nodePath]

Example:

```
--> DEL●/MANAGEMENT/USER/JohnDoe
```

Node name changed

Format: RNM●[oldNodePath]●[newNodePath]

Example:

```
--> RNM●/MANAGEMENT/USER/JohnDoe●/MANAGEMENT/USER/JaneDoe
```

5.2.8. Notifications about the changes of the properties

When the value of a property is changed and the user is subscribed to the node, which the property belongs to, an asynchronous notification is generated. This is notification is called as the 'change message'. The format of such a message is very similar to the response for the 'GET' command.

Format: CHG●[nodePath].[propertyName]=[newPropertyValue]

Example:

```
--> CHG●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I5
```

A short example of how to use the subscription

In the following, an example is presented, how the subscriptions are working and how to use them. In the example there are two independent users controlling the device through two independent connections (‘Connection #1’ and ‘Connection #2’). The events in the rows occur after each other.

CONNECTION #1	--> OPEN●/ROOM/room1/XP <-- o-●/ROOM/room1/XP --> GET●/ROOM/room1/XP.DestinationConnectionStatus <-- pw●/ROOM/room1/XP.DestinationConnectionStatus=I1;I2;I3
CONNECTION #2	--> GET●/ROOM/room1/XP.DestinationConnectionStatus <-- pw●/ROOM/room1/XP.DestinationConnectionStatus=I1;I2;I3 --> SET●/ROOM/room1/XP.DestinationConnectionStatus=I3;I2;I1 <-- pw●/ROOM/room1/XP.sw=I3;I2;I1
CONNECTION #1	--> CHG●/ROOM/room1/XP.sw=I3;I2;I1 --> CLOSE●/ROOM/room1/XP <-- c-●/ROOM/room1/XP

5.2.9. Signature

For some command the response can contain multiple lines. Every line is terminated with a carriage return (Cr, ‘\r’) and line feed (Lf, ‘\n’) characters. In several cases the number of the lines in the response cannot be determined in advanced. In several cases the client is intended waiting for the whole response and also wants to be sure, that the received lines belong together and to the same command. In these cases a special feature the ‘signature’ can be used.

The signature is a four digit long hexadecimal value that can be optionally placed before every command. In that case, the response to that particular command will also be preceded by the signature, and the corresponding lines will be in between brackets.

Command format: XXXX#[command]

Legend: xxxx: 4-digit long hexadecimal value.

Response format:

{XXXX

[command lines]

}

Legend: xxxx: 4-digit long hexadecimal value.

Example:

```
--> 0001#GET●/ROOM/roomName/XP.*
<-- {0001
<-- pw●/ROOM/roomName/XP.DestinationConnectionStatus=I1,I2,I5
<-- m-●/ROOM/roomName/XP:switch
<-- m-●/ROOM/roomName/XP:switchMulti
<-- m-●/ROOM/roomName/XP:mute
<-- }
```

Info: The lines of the signature are also Cr and Lf terminated.

5.2.10. Manual

For every node, property and method in the tree there is a manual. The manual is a human readable text that describes the syntax and provides a hint for how to use the primitives.

Command format:

for nodes: MAN●[nodePath]

for property: MAN●[nodePath].[propertyName]

for method: MAN●[nodePath]:[methodName]

Response format:

The human readable manual is separated by a space (' ') character from the primitives.

for nodes: nm●[nodePath]●Human readable manual

for properties: pm●[nodePath].[propertyName]●Human readable manual

for methods: mm●[nodePath]:[methodName]●Human readable manual

Example:

--> MAN●/MANAGEMENT/LOG

<-- nm●/MANAGEMENT/LOG This node contains different logs: warning, errors, failures.

--> MAN●/.DeviceName

<-- pm●/.DeviceName Read-only property. It contains the product name of the device.

5.2.11. Formal definitions

Method parameters and property values are specified in a modified version of Backus Naur Form (BNF). The syntax is the following:

"literal"	literals are quoted
<expression1> | <expression2>	vertical bars denote alternatives
[<expression>]	expressions in square brackets are optional
<number>*[<expression>]	expression is repeated at least <number> times
*[<expression>]	<number> may be omitted, in this case number defaults to 0
<number>*{ <expression> }	expressions in curly brackets are repeated exactly <number> times

5.2.12. Source and destination identifiers

Identifier format:

port_type = "I" | "O"

source = [port_type] decimal_number

destination = [port_type] decimal_number

Port numbers start from 1. The upper limit depends on router type, room and media layer. Port number zero ("0") serves a special purpose. Switching a source to destination zero disconnects all its destinations. Switching a destination to source zero disconnects all its sources. Some switching methods accept list of ports. In these cases port zero may not be used in conjunction with any other port. In connection status properties zero indicates that the specific source or destination is disconnected.

Each port may support one or more of the following layers: VIDEO, FORWARDAUDIO, RETURNAUDIO, RS232, INFRA and USBKVM. Port type always indicates the direction of the video signal. “I” is for input, “O” is for output. In case of switching unidirectional layers port type may be omitted. These layers are VIDEO, FORWARDAUDIO, RETURNAUDIO and USBKVM. RS232 and INFRA layers allow bidirectional routing.

E.g. “I33” is a valid destination on the RS232 layer. In this case port type is mandatory.

5.3. Examples and samples

5.3.1. Switching Methods

The following node contains the methods for switching the layers together:

/ROOM/roomName/XP

Because of different nature of layers, some connections may not be established on every layer. For instance USBKVM layer supports multipoint-to-multipoint routing but VIDEO layer is limited to point-to-multipoint routing. Switching methods will establish as much connections as possible.

The following node contains the methods for switching the layers separately:

/ROOM/roomName/XP/layerName

LayerName may be one of the following: VIDEO, FORWARDAUDIO, RETURNAUDIO, RS232, INFRA or USBKVM.

Switch one source to one destination

Method

switch

Description

Switch single source to destination in the room. This method takes a source identifier followed by a colon and a single destination. The source will be connected to the destination. All previously connected sources get disconnected from the destination. To disconnect the destination from all its sources set source to “0”.

Parameter format

parameter = source “:” destinaton

Parameter example

I1:O3

Full command example

```
--> CALL●/ROOM/example/XP/VIDEO:switch(I1:O3)
<-- mO●/ROOM/example/XP/VIDEO:switch
```

Explanation

Source 1 gets connected to Destination 3. All previously connected sources get disconnected form Destination 3.

Switch one source to all destinations

Method

switchAll

Description

Switch single source to all destinations in the room. This method connects a single source to every destination in the room. All other connections are removed. To disconnect all sources and destinations in the room set source to "0".

Parameter format

parameter = source

Parameter example

I2

Full command example

```
--> CALL●/ROOM/example/XP/VIDEO:switchAll(I2)
<-- mO●/ROOM/example/XP/VIDEO:switchAll
```

Explanation:

Source 2 gets connected to every destination in the room. All other sources get disconnected.

Switch multiple sources to multiple destinations

Method

switchMulti

Description

Switch multiple sources to multiple destinations in the room. The method takes a comma separated list of sources for each destination in the room. The listed sources will be connected to the specific destination. If the list is empty (see the example), the connections of that destination will not be changed. The first source list is assigned to the first destination, the second one is to the second, and so on. Source lists are semicolon separated.

Info: This method is efficient in reconfiguring all the connections in the room because it eliminates the overhead of explicitly specifying the destinations.

This method can process the value of DestinationConnectionStatus property directly. See DestinationConnectionStatus property for details.

Parameter format

source_list = [source] *["," source]

parameter = number_of_destinations*{ source_list ";" }

Parameter example

I1;I3,I5;;;;0;

Info: Source_list *may be empty and trailing semicolon(s) may be omitted.*

Full command example

```
--> CALL●/ROOM/example/XP/USBKVM:switchMulti(I1;I3,I5;;;;0)
<-- mO●/ROOM/example/XP/USBKVM:switchMulti
```

Explanation

The following figure shows the effects of the command:

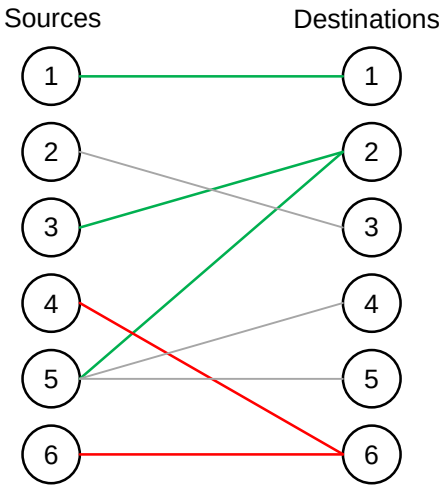


Figure 5-2. Multiple connections – example

Destination 1 gets connected to Source 1. Destination 2 gets connected to Source 3 and 5. Connections to Destination 3, 4 and 5 remain unchanged. Destination 6 gets disconnected from all its sources.

5.3.2. Modify multiple connections

Method

modifyConnections

Description

Add or remove multiple connections in the room. Each request is defined by a source identifier followed by a colon and a comma separated list of destinations. The source will be connected to the listed destinations. All other connections remain unchanged. Source may be set to “0” to disconnect the listed destinations from all their sources. Destination may be set to “0” to disconnect the specified source from all its destinations.

The method can take multiple semicolon separated requests. Requests are evaluated from left to right. Changes are preformed on a temporary buffer first. Subsequent requests may overwrite the effects of preceding ones. After processing the last request, changes are preformed in an atomic fashion.

Parameter format

destination_list = destination *[“,” destination]

parameter = 1*[source “:” destinaton_list “;”]

Parameter example

I1:O2;I4:O4,O6;I3:0;

Info: The trailing semicolon may be omitted.

Full command example

```
--> CALL●/ROOM/example/XP/VIDEO:modifyConnections(I1:O2;I4:O4,O6;I3:0)
<-- mO●/ROOM/example/XP/VIDEO:modifyConnections
```

Explanation

The following figure shows the effects of the command:

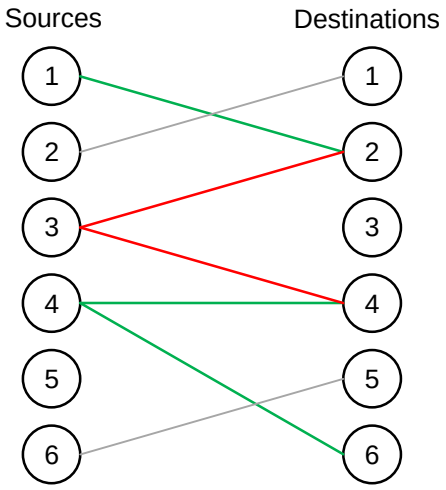


Figure 5-3. Multiple connections – example

Source 1 gets connected to Destination 2. Source 4 gets connected to Destination 4 and 6. Source 3 gets disconnected from all its destinations. Other connections remain unchanged.

5.3.3. Disconnect multiple sources from multiple destinations

Method

disconnect

Description

Disconnect multiple sources from multiple destinations in the room. Each disconnect request is defined by a source identifier followed by a colon and a comma separated list of destinations. The source will be disconnected from the listed destinations. All other connections remain unchanged. The method can take multiple requests. Requests should be separated by semicolons. For this method port number zero is not a valid source or destination.

Parameter format

destination_list = destination *["," destination]

parameter = 1*[source ":" destinaton_list ";"]

Parameter example

I3:O1;I1:O3,O4;

Info: The trailing semicolon may be omitted.

Full command example

```
--> CALL●/ROOM/example/XP/VIDEO:disconnect(I3:O1;I1:O3,O4)
<-- mO●/ROOM/example/XP/VIDEO:disconnect
```

Explanation

Source 3 gets disconnected from Destination 1. Source 1 gets disconnected from Destination 3 and 4. Other connections remain unchanged.

5.3.4. View connections on all destinations

Property

DestinationConnectionStatus

Description

View connections on all destinations in the specified room. This property gives a comma separated list of sources for each destination in the room. The first source list is assigned to the first destination, the second one is to the second, and so on. Source lists are semicolon separated.

Property value format

source_list = source *[“,” source]

property_value = number_of_destinations*{ source_list “;” }

Property value example

I1;I1,I3;I1,I6;0;0;0;

Full command example

```
--> GET●/ROOM/example/XP/USBKVM.DestinationConnectionStatus

<-- pw●/ROOM/example/XP/USBKVM.DestinationConnectionStatus=I1;I1,I3;I1,I6;
0;0;0;
```

Explanation

Source 1 is connected to Destination 1. Source 1 and 3 is connected to Destination 2. Source 1 and 6 is connected to Destination 3. Destination 4, 5 and 6 are disconnected.

5.3.5. View connections on all sources

Property

SourceConnectionStatus

Description

View connections on all sources in the specified room. This property gives a comma separated list of destinations for each source in the room. The first destination list is assigned to the first source, the second one is to the second, and so on. Destination lists are semicolon separated.

Property value format

destination_list = destination *[“,” destination]

property_value = number_of_sources*{ destination_list “;” }

Property value example

O1,O2,O3;0;O2;0;0;O3;

Full command example

```
--> GET●/ROOM/example/XP/USBKVM.SourceConnectionStatus

<-- pw●/ROOM/example/XP/USBKVM.SourceConnectionStatus=O1,O2,O3;0;O2;
0;0;O3;
```

Explanation

Destination 1, 2 and 3 are connected to Source 1. Destination 2 is connected to Source 3. Destination 3 is connected to Source 6. Source 2, 4 and 5 are disconnected.

5.4. Port status

Port status format

port_status = "-" | hexadecimal_number

Port status is a dash ("-") if no I/O card is present on the specific port or the card does not support the specific layer. Otherwise a hexadecimal status value is returned.

Info: See Port status values in the Appendix for deatils.

5.4.1. View Port Status on all destinations

Property

DestinationPortStatus

Description

View port status for all destinations in the specified room. This property gives the port status for each destination in a semicolon separated list.

Property value format

property_value = number_of_destinations*{ port_status ";" }

Property value example

-;-;2A;-;2A;2A;

Full command example

```
--> GET●/ROOM/example/XP/VIDEO.DestinationPortStatus
<-- pw●/ROOM/example/XP/VIDEO.DestinationPortStatus=-;-;2A;-;2A;2A;
```

Explanation

There are no I/O cards installed on Destination 1, 2 and 4. Destination 3, 5 and 6 is sending back status 0x2A (0x2A means no HDCP encryption, no video signal, no physical connection).

5.4.2. View Port Status on all sources

Property

SourcePortStatus

Description

View port status for all sources in the specified room. This property gives the port status for each source in a semicolon separated list.

Property value format

property_value = number_of_sources*{ port_status ";" }

Property value example

2A;2B;-;-;-;2B;

Full command example

```
--> GET●/ROOM/example/XP/VIDEO.SourcePortStatus
<-- pw●/ROOM/example/XP/VIDEO.SourcePortStatus=2A;2B;-;-;-;2B;
```

Explanation

Source 1, 2 and 6 is sending back is sending back status 0x2A (0x2A means no HDCP encryption, no video signal, no physical connection). There are no I/O cards installed on Source 3, 4 and 5.

5.5. LW3 tree structure and reference

5.5.1. / (root element)

Description: Root element

READ&WRITE PROPERTIES

- **/.Description**

  Format: .Description = <description>

  Description: Short freely editable description about the router. It can be used to easily distinguish different frames on the network.

READ-ONLY PROPERTIES

- **/.Serial**

  Format: .Serial = <serial_number>

  Description: The serial number of the router.

- **/.Name**

  Format: .Name = "25G-FR160" | "25G-FR80"

  Description: The name of the frame type. Possible values are 25G-FR160 and 25G-FR80

- **/.ProductCode**

  Format: .ProductCode = <product_code>

  Description: The product code of the router. This number helps to identify the exact type of the hardware if any support is needed.

5.5.2. /MANAGEMENT/

Description: A group node for every management related settings. The features and settings here are not related to the current crosspoint settings.

5.5.3. /MANAGEMENT/CONFIG/

Description: The node's functions are not implemented yet.

5.5.4. /MANAGEMENT/CONTROL/

Description: When using external controllers it is important to set up the appropriate connecting methods. The controllers may use different kind of protocols and they can control only just specific rooms.

In this node you can add/modify your controllers, select the appropriate protocol and adjust the TCP/IP or RS232 parameters.

At this moment only LW3 (Lightware 3), LW2 (Lightware 2) and P#2 protocols are supported. In the future new protocols may be added by firmware upgrade.

METHODS

- **/MANAGEMENT/CONTROL:create()**

  Format: :create(<controller_name>)

  Description: This method adds a new controller with the given name. All parameters will be unset, so there will be no real effect until you set them.

- **/MANAGEMENT/CONTROL:delete()**

  Format: :delete(<controller_name>)

  Description: The given controller will be deleted. The subnode will be removed.

5.5.5. /MANAGEMENT/CONTROL/<name_of_controller>/

Description: This node represents a controller. If the controller is enabled, it can access the system over the TCP/IP network. If two CPUs are running, the controller can connect to any CPU (or to both CPUs at same time), the commands will be performed regardless of which one is the active.

Each protocol has a predefined port which cannot be changed:

Port	Service/protocol
6107	Lightware 3 protocol
6207	Lightware 3 protocol over SSL (experimental)
10001	Lightware 2 protocol
1123	P#2 protocol

READ&WRITE PROPERTIES

- **/MANAGEMENT/CONTROL/<name_of_controller>.ClientIp**

  Format: .ClientIp = *[<IP_address> | <IP_range> ";"]

  Description: You can define IP addresses or ranges here. Controller connections will be accepted only from the listed IPs, so this field must be set accordingly.

  You can use the '*' wildchar instead of any octets, so you can easily cover large ranges of addresses. E.g. 192.168.0.* will accept connections from the whole subnet, while *.*.*.* will accept any connections from the network.

  You can list more IP address or range separated by a semicolon.

  Example:

  192.168.2.113;192.168.2.118;192.168.0.*

  Note: Please note that the subnet mask must be set in the /MANAGEMENT/LAN node appropriately.

  If there are more controllers, pay special attention to the IP addresses and ranges not to overlap each other.

- **/MANAGEMENT/CONTROL/<name_of_controller>.Enabled**

  Format: .Enabled = "true" | "false"

  Description: A controller can be disabled here.

  Note: Alive connections are going to not interrupted when this property is set to disabled. This setting is applicable only for new connections.

- **/MANAGEMENT/CONTROL/<name_of_controller>.Protocol**

  Format: .Protocol = "LW2" | "P#2"

  Description: You can set up the desired protocol here. At the moment only LW2 and P#2 protocols are supported.

  Note: Lightware 3 protocol is always active on port 6107 and cannot be disabled. As Lightware 3 protocol is able to handle user management, there is no need to set up it here.

- **/MANAGEMENT/CONTROL/<name_of_controller>.Room**

  Format: .Room = <name_of_room>

  Description: The room can be defined, where the controller operates. The controller will see a router with the same size as the room and it will not be able to switch outside.

  However - of course - it is possible to define the ROUTER room name. In this case controller can manage the whole crosspoint.

  Note: Only existing room names are accepted. If the room is deleted, then new connections cannot be established.

5.5.6. /MANAGEMENT/CONTROL/SERIAL/

Description: This node represents the serial ports of the CPU (which can be accessed from the black plate of the router directly). The purpose of this node is setup these ports. This node cannot be deleted.

The behavior of the RS232 ports are always the same on both CPU boards. If both CPU boards are present and working, any of them could be used, regardless of which one has the active role.

Protocol, Room and Enabled properties are operating on the same way as in other controllers.

See also: /MANAGEMENT/CONTROL/controllername

READ&WRITE PROPERTIES

- **/MANAGEMENT/CONTROL/SERIAL.Baudrate**

  Format: .Baudrate = 200 | 300 | 600 | 1200 | 1800 | 2400 | 4800 | 9600 | 19200 | 38400 | 57600 | 115200

  Description: The applied serial baud rate.

  Note: The parity bit is always turned off, the stop bit is 1.

- **/MANAGEMENT/CONTROL/SERIAL.User**

  Format: .User = <user_name>

  Description: Because of the serial port is not a session based connection, user authentication is always disabled, even if LW3 protocol is selected. (e.g. there will be never ask for login name and password on serial port). Therefore the user must be specified here. This field holds "root" as default, which means that every setting can be accessed from the serial port.

  Note: This property takes place only if LW3 is applied.

- **/MANAGEMENT/CONTROL/SERIAL.Protocol**

  See also: /MANAGEMENT/CONTROL/<name_of_controller>

- **/MANAGEMENT/CONTROL/SERIAL.Room**

  See also: /MANAGEMENT/CONTROL/<name_of_controller>

- **/MANAGEMENT/CONTROL/SERIAL.Enabled**

  See also: /MANAGEMENT/CONTROL/<name_of_controller>

5.5.7. /MANAGEMENT/DATETIME/

Description: The node allows the user to read or set the system time.

METHODS

- **/MANAGEMENT/DATETIME:setTime()**

  Format: :setTime(YYYY-MM-DDThh:mm:ss)

  Description: The system time can be set through this method. The method's argument must be formatted as an ISO date-time value.

READ-ONLY PROPERTIES

- **/MANAGEMENT/DATETIME.CurrentTime**

  Format: .CurrentTime = YYYY-MM-DDThh:mm:ss

  Description: The property shows the current system time in ISO date-time format.

5.5.8. /MANAGEMENT/EDID/

Description: The node and its subnodes represent all information about the EDID management.

Three kinds of EDIDs exist: factory, dynamic and user. Factory EDIDs are preprogrammed and cannot be changed. They are designed to cover the most possible applications. The dynamic EDIDs are read from the output ports (from the attached devices, e.g. displays or AV receivers). If a dynamic EDID is connected to an input port, then the EDID will be automatically refreshed on the inputs if a new device is connected to the output ports. User EDIDs are freely changeable - the data can be uploaded, deleted or modified with the appropriate user privileges at any time.

The EDIDs are represented by a character (F, D or U, see above) and a three-digit number starting from 001.

METHODS

- **/MANAGEMENT/EDID:save()**

Format: :save("U" <EDID_number> "," {"U" | "F" | "D"} <EDID_number>)

Description: The EDID (specified in the second parameter) is saved to the user memory (specified in the first parameter).

Example:

/MANAGEMENT/EDID:save(U13,D4)

The 4th dynamic EDID (the EDID of the device attached to the 4th output port) is saved to the 13th user memory.

Note: If the affected user memory is emulated on any input, the emulated EDIDs will be updated and the HPD line (Hot Plug Detect) will be asserted.

READ-ONLY PROPERTIES

- **/MANAGEMENT/EDID.EDIDLocations**

Format: .EDIDLocations = <frame_size>*{{"F" | "U" | "D"} <EDID_number> ";"}

Description: This property contains the summary of the currently emulated EDIDs in a semicolon separated list. The number of items matches the size of the frame (e.g. 160 at MX-FR160 or 80 at MX-FR80), where the first item represents the first input and so on.

Example:

F49;F49;F49;F49;F49;F49;F49;F49;F49;F49;F49;F49;F49;F49;.....F49;

Every input use the 49th factory EDID. This EDID is the so-called universal EDID and this is also the default.

Note: The EDID emulation can be changed from the /ROOM/room_name/EDID node.

5.5.9. /MANAGEMENT/EDID/D/

Description: This node holds the dynamic EDIDs. The EDIDs are read out from the displays by the output boards are called dynamic EDIDs.

It is important, that dynamic EDIDs are stored, so they are also present after removing the display devices.

Dynamic EDIDs are run from D001 to D160.

5.5.10. /MANAGEMENT/EDID/D/D<EDID_number>/

Description: This node represents a dynamic EDID.

Dynamic EDIDs are run from D001 to D160.

See also: /EDID/F/*

5.5.11. /MANAGEMENT/EDID/F/

Description: This node holds the factory EDIDs. The factory EDIDs are read only and their purpose to cover the most common scenarios.

Factory EDIDs are run from F001 to F097.

The available factory EDIDs at this moment:

ID	EDID name	ID	EDID name
1	LWR 640x480@60.0Hz D640x480p60	50	LWR 720x480@30.1Hz A720x480i59
2	LWR 848x480@60.0Hz D848x480p60	51	LWR 720x576@25.3Hz A720x576i50
3	LWR 800x600@60.30Hz D800x600p60	52	LWR 640x480@60.0Hz A640x480p60
4	LWR 1024x768@60.0Hz D1024x768p60	53	LWR 640x480@75.0Hz A640x480@75
5	LWR 1280x768@50.0Hz D1280x768p50	54	LWR 800x600@50.0Hz A800x600@50
6	LWR 1280x768@59.92Hz D1280x768p60	55	LWR 800x600@60.30Hz A800x600@60
7	LWR 1280x768@75.0Hz D1280x768p75	56	LWR 800x600@74.99Hz A800x600@75
8	LWR 1360x768@60.1Hz D1360x768p60	57	LWR 1024x768@49.98Hz A1024x768@50
9	LWR 1280x1024@50.0Hz D1280x1024p50	58	LWR 1024x768@60.0Hz A1024x768@60
10	LWR 1280x1024@60.1Hz D1280x1024p60	59	LWR 1024x768@75.2Hz A1024x768@75
11	LWR 1280x1024@75.1Hz D1280x1024p75	60	LWR 1280x768@50.0Hz A1280x768@50
12	LWR 1400x1050@49.99Hz D1400x1050p50	61	LWR 1280x768@59.92Hz A1280x768@60
13	LWR 1400x1050@59.99Hz D1400x1050p60	62	LWR 1280x768@75.0Hz A1280x768@75
14	LWR 1400x1050@75.0Hz D1400x1050p75	63	LWR 1360x768@60.1Hz A1360x768@60
15	LWR 1680x1050@59.99Hz D1680x1050p60	64	LWR 1364x768@50.0Hz A1364x768@50
16	LWR 1920x1080@50.0Hz D1920x1080p50	65	LWR 1364x768@59.93Hz A1364x768@60
17	LWR 1920x1080@60.0Hz D1920x1080p60	66	LWR 1364x768@74.98Hz A1364x768@75
18	LWR 2048x1080@50.0Hz D2048x1080p50	67	LWR 1280x1024@50.0Hz A1280x1024@50
19	LWR 2048x1080@59.99Hz D2048x1080p60	68	LWR 1280x1024@60.1Hz A1280x1024@60
20	LWR 1600x1200@50.0Hz D1600x1200p50	69	LWR 1366x1024@59.99Hz A1366x1024@60
21	LWR 1600x1200@60.0Hz D1600x1200p60	70	LWR 1400x1050@49.99Hz A1400x1050@50
22	LWR 1920x1200@50.0Hz D1920x1200p50	71	LWR 1400x1050@59.99Hz A1400x1050@60
23	LWR 1920x1200@59.55Hz D1920x1200p60	72	LWR 1400x1050@75.0Hz A1400x1050@75
24	LWR 2048x1200@59.95Hz D2048x1200p60	73	LWR 1920x540@50.0Hz A1920x1080i50
25	invalid	74	LWR 1920x540@59.98Hz A1920x1080i59
26	invalid	75	LWR 1920x1080@50.0Hz A1920x1080@50
27	invalid	76	LWR 1920x1080@60.0Hz A1920x1080@60
28	invalid	77	LWR 1600x1200@50.0Hz A1600x1200@50
29	LWR 1920x1080@60.0Hz Univ_DVI	78	LWR 1600x1200@60.0Hz A1600x1200@60
30	LWR 1440x240@60.3Hz H1440x480i59 2chLPCM	79	LWR 1920x1200@59.55Hz A1920x1200@60
31	LWR 1440x288@50.6Hz H1440x576i50 2chLPCM	80	LWR 1920x1200@50.0Hz A1920x1200@50
32	LWR 640x480@59.94Hz H640x480p59 2chLPCM	81	invalid
33	LWR 720x480@59.92Hz H720x480p59 2chLPCM	82	invalid
34	LWR 720x576@50.0Hz H720x576p50 2chLPCM	83	invalid
35	LWR 1280x720@50.0Hz H1280x720p50 2chLPCM	84	invalid
36	LWR 1280x720@60.0Hz H1280x720p60 2chLPCM	85	invalid
37	LWR 1920x540@50.3Hz H1920x1080i50 2chLPCM	86	invalid
38	LWR 1920x540@50.0Hz H1920x1080i50 2chLPCM	87	invalid
39	LWR 1920x540@59.98Hz H1920x1080i59 2chLPCM	88	invalid
40	LWR 1920x540@60.5Hz H1920x1080i60 2chLPCM	89	LWR 1920x1200@59.55Hz Univ_Analog
41	LWR 1920x1080@24.0Hz H1920x1080p24 2chLPCM	90	LWR 1920x2160@59.98Hz D1920x2160@60
42	LWR 1920x1080@24.99Hz H1920x1080p25 2chLPCM	91	LWR 1024x2400@60.1Hz D1024x2400@60
43	LWR 1920x1080@30.0Hz H1920x1080p30 2chLPCM	92	LWR 1920x2400@59.97Hz D1920x2400@60
44	LWR 1920x1080@50.0Hz H1920x1080p50 2chLPCM	93	LWR 2048x2400@59.97Hz D2048x2400@60
45	LWR 1920x1080@59.93Hz H1920x1080p59 2chLPCM	94	LWR 2048x1536@59.99Hz D2048x1536@60
46	LWR 1920x1080@60.0Hz H1920x1080p60 2chLPCM	95	LWR 2048x1536@74.99Hz D2048x1536@75
47	LWR 1920x1080@60.0Hz Univ_HDMI_PCM 2chLPCM	96	LWR 2560x1600@59.85Hz D2560x1600@60
48	LWR 1920x1080@60.0Hz Univ_HDMI_ALL 2chLPCM,4chLPCM,DD,DTS,AAC,DD+,DTS-HD,MLP,DST,WMAPro	97	LWR 3840x2400@23.99Hz D3840x2400@24
49	LWR 1920x1080@60.0Hz Univ_HDMI_DC 2chLPCM,4chLPCM,DD,DTS,AAC,DD+,DTS-HD,MLP,DST,WMAPro		

5.5.12. /MANAGEMENT/EDID/F/F<EDID_number>/

Description: This node represents an EDID.

READ&WRITE PROPERTIES

- **/MANAGEMENT/EDID/F/F<EDID_number>.data**

Format: .data = 256*{<2_octet_hex>}

Description: The raw data of the EDID. This is in binary format. Binary formats are represented as HEXA ASCII values in the protocol, such as:

00FFFFFFFFFF0032F200..00C9 (512 character)

Note: This property is not writable if the EDID is factory or dynamic.

READ-ONLY PROPERTIES

- **/MANAGEMENT/EDID/F/F<EDID_number>.header**

Format: .header = <EDID_description>

Description: This field contains the auto-generated short summary of the EDID.

This field holds the MFID (3 characters), the preferred detailed timing in short form (e.g. 640x480@60.0Hz), the name of the EDID extracted from the Name descriptor and the list of supported audio formats.

Example:

LWR 1920x1080@60.0Hz Univ_HDMI_ALL 2chLPCM,4chLPCM,DD,DTS,AAC,DD+, DTS-HD,MLP,DST,WMAPro

Note: If the EDID is not valid, this property holds the "invalid" string.

5.5.13. /MANAGEMENT/EDID/U/

Description: This node contains user saved EDIDs. The number of user EDIDs are 100 and these memory slots are empty at default.

User saved EDIDs are run from U001 to U100.

5.5.14. /MANAGEMENT/EDID/U/U<EDID_number>/

Description: This node represents a user memory EDID.

User memories run from U001 to U100.

See also: /EDID/F/*

5.5.15. /MANAGEMENT/GENLOCK/

Description: This node will contain the genlock settings.

5.5.16. /MANAGEMENT/INFRA/

Description: The infra layer related global settings are accessible here.

METHODS

- **/MANAGEMENT/INFRA:DeviceList()**

Format: :DeviceList(*[<manufacturer> "/" <device> ";"])

Description: List of devices which are used in the system. The infra codes will be processed using the listed devices' code list.

5.5.17. /MANAGEMENT/INFRA/DATABASE/

Description: Infra code database

METHODS

- **/MANAGEMENT/INFRA/DATABASE:add()**

  Format: :add(<manufacturer_name>)

  Description: A new manufacturer can be added to the database with no devices.

- **/MANAGEMENT/INFRA/DATABASE:delete()**

  Format: :delete(<manufacturer_name>)

  Description: A manufacturer can be deleted from the database. The manufacturer must have no devices in the database or the deletion will fail.

5.5.18. /MANAGEMENT/INFRA/DATABASE/<manufacturer>/

Description: The list of available devices for a manufacturer (the node name is the capitalized name of the manufacturer, e.g. SAMSUNG, SONY, etc...).

METHODS

- **/MANAGEMENT/INFRA/DATABASE/<manufacturer>:add()**

  Format: :add(<device_name>)

  Description: A new device can be added to the database. The new devices will contain no codes.

- **/MANAGEMENT/INFRA/DATABASE/<manufacturer>:delete()**

  Format: :delete(<device_name>)

  Description: A device can be deleted from this manufacturer. The device must have no codes or the deletion will fail.

5.5.19. /MANAGEMENT/INFRA/DATABASE/<manufacturer>/<device>/

Description: The list of available infra codes for a given device. The node name is the capitalized name of the device, e.g. TX-SR605, BRAVIATV, ...

METHODS

- **/MANAGEMENT/INFRA/DATABASE/<manufacturer>/<device>:add()**

  Format: :add(<function_name> ";" *[<4_octet_hex> " "])

  Description: It adds a new infra code to the database. The first parameter is the function name (a new property will be added with that name), while the second parameter is a pronto hex code. The pronto hex code must be valid and it can contain two different burst sequence.

  Example:

  CALL /MANAGEMENT/INFRA/DATABASE/TEST/SONY/BD:add(PLAY;0000 006D 000C 0000 0022 0021 0022 0022 0046 0040 0047 001F 0025 001E 0022 0023 0022 0021 0022 0022 0021 0022 0022 0021 0022 0043 0023 0720)

- **/MANAGEMENT/INFRA/DATABASE/<manufacturer>/<device>:delete()**

  Format: :delete(<layer_id>)

  Description: An infra code can be deleted but only the one added by the user previously. The parameter is the function name (same as the property name).

READ&WRITE PROPERTIES

- **/MANAGEMENT/INFRA/DATABASE/<manufacturer>/<device>.CODENAME**

  Format: .CODENAME = "FACTORY" | *[<4_octet_hex> " "]

  Description: The name of the property is the function name of the IR code and it is capitalized by convention. Example for names: REPEAT, PLAY, STOP, 1, 2, ...

  The factory shipped codes are returning with a **FACTORY** value due to legacy reasons, while the user programmed codes will return the pronto hex format.

5.5.20. /MANAGEMENT/LAN/

Description: Ethernet settings, such as IP address, subnet mask, etc.

LAN settings can be set independently for both CPUs.

5.5.21. /MANAGEMENT/LAN/CPU<number>/

Description: The LAN settings of the CPUs. The value of the number in the node name is 1 or 2 which reflects to the first or second slots.

METHODS

- **/MANAGEMENT/LAN/CPU<number>:apply()**

  Description: The method has to be called to apply the new IP settings. After calling this method, the DHCP state and the static addresses are loaded to the CPU board.

READ&WRITE PROPERTIES

- **/MANAGEMENT/LAN/CPU<number>.Dhcp**

  Format: .Dhcp = "true" | "false"

  Description: Use DHCP server for obtaining LAN settings or not. If DHCP server is applied, then every other property in this node is invalid.

  Note: Using DHCP server in a real production environment is not suggested. Use fix LAN settings whenever possible.

  The actual IP addresses are always shown on the front panel LCD.

- **/MANAGEMENT/LAN/CPU<number>.StaticIp**

  Format: .StaticIp = <ip_address>

  Description: The static IP address of the CPU board. Only IPv4 is supported. Default values are 192.168.0.101 and 192.168.0.102 for the CPU boards. If DHCP is applied, this value is not used.

- **/MANAGEMENT/LAN/CPU<number>.StaticGateway**

  Format: .StaticGateway = <ip_address>

  Description: The static default gateway address for the CPU board. Only IPv4 is supported. Default value is 192.168.0.1. If DHCP is applied, this value is not used.

- **/MANAGEMENT/LAN/CPU<number>.StaticSubnet**

  Format: .StaticSubnet = <subnet_mask>

  Description: The subnet mask for the CPU board. Only IPv4 is supported. Default value is 255.255.255.0. If DHCP is applied, this value is not used.

READ-ONLY PROPERTIES

- **/MANAGEMENT/LAN/CPU<number>.ActiveIp**

  Format: .ActiveIp = <IP_address>

Description: The active IP address of the CPU board. Only IPv4 is supported. If DHCP is used, this address is received from the DHCP server. If DHCP is turned off, this is the static IP address.

- **/MANAGEMENT/LAN/CPU<number>.ActiveSubnet**

  Format: .ActiveSubnet = <subnet_mask>

  Description: The subnet mask for the CPU boards. If DHCP is used, this address is received from the DHCP server. If DHCP is turned OFF, this is the static subnet mask.

  Note: CIDR range from 256 A to 1/256 C is accepted.

- **/MANAGEMENT/LAN/CPU<number>.ActiveGateway**

  Format: .ActiveGateway = <IP_address>

  Description: Default gateway for the CPU boards. If DHCP is used, this address is received from the DHCP server. If DHCP if turned off, this is the static default gateway.

5.5.22. /MANAGEMENT/LAN/SBC/

Description: IP settings of the SBC.

See also: /MANAGEMENT/LAN/CPU<number>

5.5.23. /MANAGEMENT/LICENSE/

Description: The node collects all the installed licenses to the router. It has also methods for adding / removing licenses.

METHODS

- **/MANAGEMENT/LICENSE:install()**

  Format: :install(<license_code>)

  Description: A new license can be installed. The license code sent by Lightware must be passed as a parameter. The two possible return values are fail and ok. If the new license is installed, then it is saved to the internal memory and a new subnode appears.

  It is important that the license codes are bound to the serial number of the frame. It is not possible to apply the same code on another router.

  Example:

  CALL /MANAGEMENT/LICENSE:install(QRZ3WC1-KTMOQO1-EY62P30)

  Note: A restart may be required to apply the new settings. See the RestartReq property.

- **/MANAGEMENT/LICENSE:remove()**

  Format: :remove(<license_code>)

  Description: It removes an existing license. The parameter must be the license code. The license will be removed permanently.

  Note: A restart may be required to apply the new settings. See the RestartReq property.

READ-ONLY PROPERTIES

- **/MANAGEMENT/LICENSE.RestartReq**

  Format: .RestartReq = "true" | "false"

  Description: If this property is true, then a restart is required to apply the new license settings (e.g. added/removed license features)

5.5.24. /MANAGEMENT/LICENSE/<license_code>/

Description: The node stores the details about an installed license option. The name of the node is same as the license code.

READ-ONLY PROPERTIES

- **/MANAGEMENT/LICENSE/<license_code>.Key**

  Format: .Key = <license_code>

  Description: The property contains the license code, which is equal to the node's name.

- **/MANAGEMENT/LICENSE/<license_code>.Name**

  Format: .Name = <description>

  Description: A brief explanation about license option. (e.g. Size 120x120, Infra layer, ...)

- **/MANAGEMENT/LICENSE/<license_code>.ProductCode**

  Format: .ProductCode = <product_code>

  Description: The product code of the license option if available, N/A in other cases.

- **/MANAGEMENT/LICENSE/<license_code>.DateOfIssue**

  Format: .DateOfIssue = <date_of_issue>

  Description: The date (dd.mm.yyyy) of the issuing this license by Lightware.

- **/MANAGEMENT/LICENSE/<license_code>.DateOfExpire**

  Format: .DateOfExpire = <date_of_expire>

  Description: The date (dd.mm.yyyy) of the expiration of this license. This license will be not active after this date.

  Usually this property returns never, which means that the license never expires.

- **/MANAGEMENT/LICENSE/<license_code>.Description**

  Format: .Description = <description>

  Description: A brief explanation about license option. (e.g. Size 120x120, Infra layer, ...)

- **/MANAGEMENT/LICENSE/<license_code>.Active**

  Format: .Active = "true" | "false"

  Description: Is this license is active? This property returns false if the license is expired.

5.5.25. /MANAGEMENT/LOG/

Description: The last logged events can be queried here. A controller can get every log message if it subscribes to this node. There is no protocol support to query the whole log - this can be done by FTP protocol.

The entries have five semicolon separated fields. The first one is the date, the second one is the source board (e.g. CPU1, IN12...), the third one is the error level, the fourth one is the short description while the fifth one is a hexadecimal parameter. Sometimes a sixth field follows with more deep information.

READ-ONLY PROPERTIES

- **/MANAGEMENT/LOG.DEBUG**

  Format: .DEBUG = <date_time> ";" <board_name> ";" <level> ";" <description> ";" <parameter> ";" [<info_text>]

  Description: The last debug message

- **/MANAGEMENT/LOG.NOTICE**

  Format: .NOTICE = <date_time> ";" <board_name> ";" <level> ";" <description> ";" <parameter> ";" [<info_text>]

  Description: The last notice level message. These are logging events which may be important but not extraordinary. For example new or terminated TCP/IP connections are logged as notice.

- **/MANAGEMENT/LOG.WARNING**

  Format: .WARNING = <date_time> ";" <board_name> ";" <level> ";" <description> ";" <parameter> ";" [<info_text>]

  Description: The last warning message. These messages may refer some extraordinary unexpected event which needs to get attention, however these issues can be handled automatically by the firmware and they do not cause any external perceptible side effect.

- **/MANAGEMENT/LOG.ERROR**

  Format: .ERROR = <date_time> ";" <board_name> ";" <level> ";" <description> ";" <parameter> ";" [<info_text>]

  Description: The last error message. Error messages needs immediately attention, they represent an issue where a part of the router (e.g. an I/O port or a whole board) is inoperable, however other parts have remained in working state.

- **/MANAGEMENT/LOG.FATAL**

  Format: .FATAL = <date_time> ";" <board_name> ";" <level> ";" <description> ";" <parameter> ";" [<info_text>]

  Description: The last fatal error message. These messages refer to issues which prevent the router to continue the operation and one or more of the main functions fail to work. For example a crosspoint chip error or multiple power issues may trigger fatal errors.

5.5.26. /MANAGEMENT/SYSTEM/

Description: This node represents the hardware. Each kind of boards have a read-only property, where the actual statuses of the boards are reported back. Every status information contains a letter which refers to the overall health report of the board and a one-digit hex number called state_id which describes the actual state.

The meaning of the first character:

state_id	Description
o	There are no extraordinary event related to the board
w	There is a warning message from the board(eg. the temperature has reached the warning level)
e	Some kind of serious hardware error has happened

The meaning of the state_id values:

state_id	Description
0	The slot is not exist, because the parent board is removed. (e.g. there is no crossboard, so I/O board cannot be inserted)
1	The slot is empty, the board is not present
2	The board is present but not powered
3	The power is applied, waiting for stable voltages
4	The board has started, booting
5	The router tries communicate with the board
6	Connection is established, initialization is running
7	The board is operational
8	The board is under shutting down
9	The board is powered down by user
A	Firmware upgrade is in progress
B	Serial firmware upgrade is in progress
C	Reset is performed,
D	The board is disabled due to license limitations

Every board in the frame has a subnode under this node. Please note that not every subnode is documented as this part of the protocol tree serves only debug purposes.

METHODS

- **/MANAGEMENT/SYSTEM:powerOn()**

  Description: The method turns on power supplies and all cards in the router. Call only if it is in standby mode.

- **/MANAGEMENT/SYSTEM:powerOff()**

  Description: The method turns off every card, shutdown the SBC and turn off the PSUs.

READ-ONLY PROPERTIES

- **/MANAGEMENT/SYSTEM.PCB**

  Format: .PCB = {{"o" | "w" | "e"} <state_id> ";"}

  Description: The state of Power control board.

  Example: o7;

  Note: Depending on the frame version, the system may have two PSB or a PCB.

- **/MANAGEMENT/SYSTEM.SB**

  Format: .SB = {"o" | "w" | "e"} <state_id> ";"

  Description: The states of the status board (below the touchscreen).

  Example: o7;

- **/MANAGEMENT/SYSTEM.CPUB**

  Format: .CPUB = <number_of_CPU_boards>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The states of CPU boards.

  Example: o7;o7;

  Note: All frames have two CPU board slots.

- **/MANAGEMENT/SYSTEM.PSU**

  Format: .PSU = <number_of_psus>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The state of the power supplies.

  Note: The 25G-FR160 has 4 PSU-s.

- **/MANAGEMENT/SYSTEM.XB**

  Format: .XB = <number_of_crosspointboards>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The states of crosspoint boards.

  Example: o0;o0;o0;o0;o0;o0;

  Note: MX-FR160 has six crosspoint boards. The first four boards are for the video, the last two boards are for the audio/usb 2.0.

- **/MANAGEMENT/SYSTEM.status**

  Format: .status = {"Booting" | "Running" | "Shutting down" | "Standby"}

  Description: The property indicates the actual system status. When the router applied to the power network, the default state will be Running.

- **/MANAGEMENT/SYSTEM.FT**

  Format: .FT = <number_of_fan_trays>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The state of the fan control boards.

  Example: o7;w2;o7;

  Note: MX-FR160 has three fan control boards.

- **/MANAGEMENT/SYSTEM.LSCB**

  Format: .LSCB = <number_of_lscb>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The states of low speed control boards.

  Example: o7;o6;e3;o1;

  Note: MX-FR160 has four crossboards.

- **/MANAGEMENT/SYSTEM.CCB**

  Format: .CCB = <number_of_ccb>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The states of communication control boards.

  Example: e1;o3;o1;

  Note: MX-FR160 has three control boards.

- **/MANAGEMENT/SYSTEM.EB**

  Format: .EB = {"o" | "w" | "e"} <state_id> ";"

  Description: The state of the Ethernet board.

  Example: o7;

  Note: MX-FR160 has one Ethernet board.

- **/MANAGEMENT/SYSTEM.XCB**

  Format: .XCB = {"o" | "w" | "e"} <state_id> ";"

  Description: The state of the Crosspoint cross board.

  Example: o7;

  Note: MX-FR160 has one Crosspoint cross board.

- **/MANAGEMENT/SYSTEM.MOB**

  Format: .MOB = {{"o" | "w" | "e"} <state_id> ";"}

  Description: The state of the Monitor board.

  Example: o7;

  Note: MX-FR160 has one Monitor board.

- **/MANAGEMENT/SYSTEM.GB**

  Format: .GB = {{"o" | "w" | "e"} <state_id> ";"}

  Description: The state of Genlock board.

  Example: o6;

  Note: MX-FR160 has one monitor board.

- **/MANAGEMENT/SYSTEM.SBCCB**

  Format: .SBCCB = {{"o" | "w" | "e"} <state_id> ";"}

  Description: The state of SBC connect board.

  Example: o7;

  Note: MX-FR160 has one SBC connect board.

- **/MANAGEMENT/SYSTEM.IN**

  Format: .IN = <number_of_input_slots>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The states of input boards.

  Example: o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;

  Note: MX-FR160 has 20 input boards. The first ten boards are for the first LSCB, the next ten boards are for the second LSCB.

- **/MANAGEMENT/SYSTEM.OUT**

  Format: .OUT = <number_of_output_slots>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The states of output boards.

  Example: o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;o0;

  Note: MX-FR160 has 20 output boards. The first 10 boards are for the third LSCB, the next 10 boards are for the fourth LSCB.

- **/MANAGEMENT/SYSTEM.PSB**

  Format: .PSB = <number_of_ccb>*{{"o" | "w" | "e"} <state_id> ";"}

  Description: The state of the Power sum boards.

  Example: o7;o7;

  Note: Depending on the frame version, the system may have two PSB or a PCB.

5.5.27. /MANAGEMENT/SYSTEM/<board_type><id>/

Description: Boards which connected directly (not over ethernet) to the CPU are using this interface. There boards are: FT, XCB, GB, PSB, XB

METHODS

- **/MANAGEMENT/SYSTEM/<board_type><id>:reset()**

  Description: The method resets the board.

- **/MANAGEMENT/SYSTEM/<board_type><id>:powerOff()**

  Description: The method turns off the board.

- **/MANAGEMENT/SYSTEM/<board_type><id>:powerOn()**

  Description: The method turns on the board if it was previously turned off.

READ-ONLY PROPERTIES

- **/MANAGEMENT/SYSTEM/<board_type><id>.State**

  Description: The status of the board as a human readable string.

5.5.28. /MANAGEMENT/SYSTEM/<board_type><id>/HEALTH/

Description: The node contains the measured values by the boards sensors.

Note: The node existance may depend on the board type.

READ-ONLY PROPERTIES

- **/MANAGEMENT/SYSTEM/<board_type><id>/HEALTH.<sensor_value>**

  Format: . = [<value> <unit> "[" <min_warning> ";" <max_warning> "] [" <min_error> ";" <max_error> "]"]

  Description: The property shows the value of the sensor, and it's warning and error limits.

5.5.29. /MANAGEMENT/SYSTEM/<board_type><id>/UID/

Description: The board's Unique Identifier fields.

Note: The node existance may depend on the board type.

READ&WRITE PROPERTIES

- **/MANAGEMENT/SYSTEM/<board_type><id>/UID.<UID_field>**

  Description: The UID data field value. The available UID fields:

  UidVersion, ProductName, ProductSerialNumber, AccessorySerialNumber, FirmwareVersion, AdditionalFirmwareVersion, HwVersion, AddonHwVersion,

ProductPartNumber, AccessoryPartNumber, ArmPresent, ArmDfuBoot, ArmEthBoot, ArmUsbBoot, ArmStandby, Tested, FailedOnTest, EngineeringSample, DemoSample, ServiceCounter, DateOfAssembly, LocationOfAssembly, DateOfTest, LocationOfTest

5.5.30. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/

Description: Boards which connected to the CPU over ethernet are using this interface. There boards are: LSCB, MOB, SBCCB, IN, OUT, SB

METHODS

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>:reset()**

  Description: The method resets the board.

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>:powerOff()**

  Description: The method turns off the board.

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>:powerOn()**

  Description: The method turns on the board if it was previously turned off.

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>:firmwareUpgradeEthernet()**

  Description: The method sets the board into firmware upgrade mode (over ethernet).

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>:firmwareUpgradeSerial()**

  Description: The method sets the board into serial firmware upgrade mode.

READ-ONLY PROPERTIES

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>.State**

  Description: The status of the board as a human readable string.

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>.Name**

  Description: The name of the board.

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>.Serial**

  Description: The serial number of the board.

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>.Fwver**

  Description: The firmware version of the board.

- **/MANAGEMENT/SYSTEM/<connectable_board_type><id>.socket_id**

  Description: The board connection's socket ID. Used for debug purposes.

5.5.31. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/HEALTH/

See also: /MANAGEMENT/SYSTEM/<board_type><id>/HEALTH

5.5.32. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/MNT/

Description: The connected board's root node is mounted under this node. It's contents are depending on the board.

5.5.33. /MANAGEMENT/SYSTEM/<connectable_board_type><id>/UID/

See also: /MANAGEMENT/SYSTEM/<board_type><id>/UID

5.5.34. /MANAGEMENT/SYSTEM/CCB<id>/

Description: The node has the general board's properties, but the Communication control board's node has a few more properties.

See also: /MANAGEMENT/SYSTEM/<board_type><id>

READ-ONLY PROPERTIES

- **/MANAGEMENT/SYSTEM/CCB<id>.<board><id>**

  Format: . = ["N/A" | "up" | "down"]

  Description: The property indicates if the ethernet link is up/down to the board specified by the property name. When the CCB is not operational, the properties are filled with N/A.

5.5.35. /MANAGEMENT/SYSTEM/CCB<id>/UID/

See also: /MANAGEMENT/SYSTEM/<board_type><id>/UID

5.5.36. /MANAGEMENT/SYSTEM/CPUB<id>/

Description: The CPU boards (The id is 1 or 2, reflecting to the first and second slots)

METHODS

- **/MANAGEMENT/SYSTEM/CPUB<id>:activate()**

  Description: This method call will change the active/backup roles if this CPU operates in backup mode. After calling the method this CPU will become the active CPU.

  Note: In case of removing or faulting of the active CPU, the backup CPU will automatically take the active role over.

- **/MANAGEMENT/SYSTEM/CPUB<id>:reset()**

  Description: The method resets the board.

READ-ONLY PROPERTIES

- **/MANAGEMENT/SYSTEM/CPUB<id>.State**

  Description: The status of the board as a human readable string.

- **/MANAGEMENT/SYSTEM/CPUB<id>.Fwver**

  Description: The firmware version of the board.

- **/MANAGEMENT/SYSTEM/CPUB<id>.Active**

  Format: .Active = "true" | "false"

  Description: Is this CPU board the active CPU? If the value is false, then this CPU operates as the backup CPU.

5.5.37. /MANAGEMENT/SYSTEM/CPUB<id>/HEALTH/

Description: The node basicly filled with the general UID node properties, but it has a few more, which are CPU specific.

See also: /MANAGEMENT/SYSTEM/<board_type><id>/HEALTH

READ-ONLY PROPERTIES

- **/MANAGEMENT/SYSTEM/CPUB<id>/HEALTH.CoreVersion**

  Format: .CoreVersion = [major].[minor].[subminor]

  Description: The version of the CPU core software.

- **/MANAGEMENT/SYSTEM/CPUB<id>/HEALTH.BuildTime**

  Description: The exact date-time when the CPU core software was built.

  Example: May 29 2014 12:10:45

- **/MANAGEMENT/SYSTEM/CPUB<id>/HEALTH.FpgaVersion**

  Description: The version of the CPU board's FPGA.

5.5.38. /MANAGEMENT/SYSTEM/CPUB<id>/UID/

See also: /MANAGEMENT/SYSTEM/<board_type><id>/UID

5.5.39. /MANAGEMENT/SYSTEM/SB/

Description: The Status board's node. It's the same as the connectable boards' node, but it doesn't have powerOn and powerOff methods.

See also: /MANAGEMENT/SYSTEM/<connectable_board_type><id>/

5.5.40. /MANAGEMENT/SYSTEM/SB/HEALTH/

See also: /MANAGEMENT/SYSTEM/<connectable_board_type><id>/HEALTH

5.5.41. /MANAGEMENT/SYSTEM/SB/MNT/

See also: /MANAGEMENT/SYSTEM/<connectable_board_type><id>/MNT

5.5.42. /MANAGEMENT/SYSTEM/SB/UID/

See also: /MANAGEMENT/SYSTEM/<connectable_board_type><id>/UID

5.5.43. /MANAGEMENT/USERS/

Description: This node controls the user management. The subnodes represents the each users, where the subnode name is equal to the case-sensitive user name.

The router can handle unlimited users. The access rights to the nodes for each users can be set up independently.

There are three access right:

Name	Description
Deny	It denies both read and write a node.
Read	It allows to read a node.
Write	It allows both read and write a node.

If an access rule is defined for a node, then it applies to all subnodes, unless there is another rule to a subnode. If there is no rule defined, then everything can be read and write.

When adding a new user, it is recommended to set the deny for the root ("/") node (this prohibits the whole tree) then add the read and write exceptions to the specific nodes.

METHODS

- **/MANAGEMENT/USERS:create()**

  Format: :create(<name_of_user>)

  Description: A new user can be created. A new node will be appear under the USERS node.

- **/MANAGEMENT/USERS:delete()**

  Format: :delete(<name_of_user>)

  Description: An existing user can be deleted.

  Note: The root user cannot be deleted.

- **/MANAGEMENT/USERS:setLoginRequired()**

  Format: :setLoginRequired({"true" | "false"} ";" <root_password>)

  Description: The user management can be enabled or disabled for the router. For security reasons the root password must be provided in the second parameter.

Example:

CALL /MANAGEMENT/USERS:setLoginRequired(true;mysecretpassword)

Note: The default root password is "admin".

READ-ONLY PROPERTIES

- **/MANAGEMENT/USERS.LoginRequired**

  Format: .LoginRequired = "true" | "false"

  Description: The value of this property is true if the user management is enabled in the router. If the user management is disabled, then the new incoming LW3 connections on the port 6107 are accepted without requiring a user name and password and all connections are treated as root session.

  If the user management is turned on, then at connecting a user name and a password will be required.

  Note: The user management can be enabled by the :setLoginRequired() method.

  The user management is disabled per default.

- **/MANAGEMENT/USERS.Active**

  Format: .Active = *[<user_name> "@" {<ip_address> | "SERIAL"} ";"]

  Description: This property lists all active sessions separated by semicolon. Every item consists the user name and the IP address separated by the @ sign. If RS232 connection is used, the "SERIAL" string will be appear instead of the IP address.

  Example:

  /MANAGEMENT/USERS.Active=root@192.168.2.73;bob@192.168.2.15;johndoe@SERIAL;

  Note: Every incoming connection will be listed regardless of the used protocol.

5.5.44. /MANAGEMENT/USERS/<name_of_user>/

Description: This node represents a user.

READ&WRITE PROPERTIES

- **/MANAGEMENT/USERS/<name_of_user>.Browse**

  Format: .Browse = "true" | "false"

  Description: If true, user can browse all nodes (query the node names, but not the properties itself).

  If it is false, then the user can browse only the nodes, where he has at least read permission.

  Note: Default value is false.

- **/MANAGEMENT/USERS/<name_of_user>.Deny**

  Format: .Deny = *[<node_path> ";"]

  Description: The list of nodes, where the read and write permissions are denied.

  Example:

  /MANAGEMENT/USERS/bob.Deny=/;/ROOM/test/PORTS;/ROOM/test/EDID

  Note: It is recommended to add the root node here for new users.

- **/MANAGEMENT/USERS/<name_of_user>.Enabled**

  Format: .Enabled = "true" | "false"

  Description: A user can be disabled here. If this field is false, the user can't log on.

  Note: Default value is true.

- **/MANAGEMENT/USERS/<name_of_user>.Name**

  Format: .Name = <name_of_the_user>

  Description: A freely editable field, where e.g. the user real name can be noted.

  Note: Avoid non-English letters.

- **/MANAGEMENT/USERS/<name_of_user>.Password**

  Format: .Password = <password>

  Description: The password of the user.

  Note: Password is case sensitive.

- **/MANAGEMENT/USERS/<name_of_user>.Read**

  Format: .Read = *[<node_path> ";"]

  Description: The list of nodes, where read permissions are granted.

  Note: All subnodes are affected unless no exception is defined.

- **/MANAGEMENT/USERS/<name_of_user>.Write**

  Format: .Write = *[<node_path> ";"]

  Description: The list of nodes, where write permissions are granted.

  Note: All subnodes are affected unless no exception is defined.

5.5.45. /MANAGEMENT/USERS/root/

Description: This user is always present. It has write rights to every node and that cannot be changed.

See also: /MANAGEMENT/USERS/<name_of_user>

5.5.46. /MANAGEMENT/VCP/

Description: Virtual Control Ports can be accessed from here.

The purpose of the control ports is sending (injecting) or receiving (capturing) data to/from the remote serial and infra ports. VCPs can be added to the rooms and they can be switched like other real ports.

5.5.47. /MANAGEMENT/VCP/INFRA/

Description: Infrared virtual control ports.

The node's existance may depend on the applied licenses.

5.5.48. /MANAGEMENT/VCP/INFRA/V<port_number>/

Description: A Virtual Control Port for infrared data. The port number is between 1 and 32.

METHODS

- **/MANAGEMENT/VCP/INFRA/V<port_number>:send()**

  Format: :send(<manufacturer> "/" <device> "." <function>)

  Description: It sends an infra code stored in the database. The parameter must be the database path to the infra code.

  Example:

  CALL /MANAGEMENT/VCP/INFRA/V1.send(/HITACHI/VCR.POWER)

  Note: Every codes can be sent from the database anytime, regardless of whether the /MANAGEMENT/INFRA.DeviceList property lists the actual device or not.

- **/MANAGEMENT/VCP/INFRA/V<port_number>:sendRaw()**

  Format: :sendRaw(*[<4_digit_hex> " "])

  Description: Sends a user-provided infra code. Parameter must be in pronto hex format.

  Example:

  CALL /MANAGEMENT/VCP/INFRA/V1.sendRaw(0000 006D 000C 0000 0022 0021 0022 0022 0046 0040 0047 001F 0025 001E 0022 0023 0022 0021 0022 0022 0021 0022 0022 0021 0022 0043 0023 0720)

- **/MANAGEMENT/VCP/INFRA/V<port_number>:learn()**

  Description: If this method is called, then next incoming infra code will be not searched against the database, but will be returned in the Rx property in pronto hex format.

  If you want learn a new infra code, you have to call the learn function, press the button, and get the pronto hex code from the CHG message. This pronto hex code can be added to the database after then.

  Note: This method is only for learning one code. After returning a protno hex code, the VCP port reverts back to the normal mode.

  READ-ONLY PROPERTIES

- **/MANAGEMENT/VCP/INFRA/V<port_number>.Pid**

  Format: .Pid = 0x<PID>

- **/MANAGEMENT/VCP/INFRA/V<port_number>.Rx**

  Format: .Rx = {{ <manufacturer> "/" <device> "." <function>} | { *[<4_digit_hex> " "]}

  Description: This property represents the actually received data. When reading this property by GET command, it will always return with an empty string, however after subscribing to the VCP node (by the OPEN command) a CHG will be sent after every decoded infra command.

  The incoming commands will be compared against the codes of the devices which are listed in the /MANAGEMENT/INFRA.DeviceList property. If the code is recognized, then a CHG message will be fired with the database path of the code. Unrecognized codes will not trigger any messages.

  Example:

  CHG /MANAGEMENT/VCP/INFRA/V1.Rx=/HITACHI/VCR.POWER

  If the learn method has been called before, then the incoming data will be not compared against the database, but the Rx will return with the received code in pronto hex format.

  Example:

  CHG /MANAGEMENT/VCP/INFRA/V1.Rx=0000 006D 000C 0000 0022 0021 0022 0022 0046 0040 0047 001F 0025 001E 0022 0023 0022 0021 0022 0022 0021 0022 0022 0021 0022 0043 0023 0720

  Note: The router decodes and interprets internally the common infra code protocols during the database search, therefore the codes will be detected regardless of the actual state of the toggle bits.

5.5.49. /MANAGEMENT/VCP/INFRA/V<port_number>/DATABASE/

Description: The /INFRA/DATABASE node is mounted here.

See also: /INFRA/DATABASE

5.5.50. /MANAGEMENT/VCP/KVM/

Description: USB KVM virtual control ports.

The node's existance may depend on the applied licenses.

5.5.51. /MANAGEMENT/VCP/KVM/V<port_number>/

READ-ONLY PROPERTIES

- **/MANAGEMENT/VCP/KVM/V<port_number>.Pid**

  Format: .Pid = 0x<PID>

  Description: The VCP's internal identifier.

5.5.52. /MANAGEMENT/VCP/RS232/

Description: Serial virtual control ports.

The node's existance may depend on the applied licenses.

5.5.53. /MANAGEMENT/VCP/RS232/V<port_number>/

Description: A Virtual Control Port for serial data. The port number is between 1 and 32.

METHODS

- **/MANAGEMENT/VCP/RS232/V<port_number>:Tx()**

  Format: :Tx(<text_to_send>)

  Description: This method sends data to the serial VCP port. The maximum length of the text is 128 byte, longer messages needs to be split. Every remote serial ports have a 128-byte long FIFO, the controller must take care about the amount of sent text. If too much data is sent in a short period, the FIFO may overflow and data will be lost.

READ-ONLY PROPERTIES

- **/MANAGEMENT/VCP/RS232/V<port_number>.Pid**

  Format: .Pid = 0x<PID>

  Description: The VCP's internal identifier.

- **/MANAGEMENT/VCP/RS232/V<port_number>.Rx**

  Format: .Rx = <received_text>

  Description: Received data on the port. When this property is queried by GET command, it is always empty, but the subscribers get CHG responses with the received text when new data is available.

  Note: The text is escaped according to the protocol description. Long incoming messages will be split into more parts.

5.5.54. /ROOM/

Description: The ports of the matrix can be organized in rooms, which allows the administrator to grant access for a user only for a group of ports. This is useful for security reasons and also simplifies the life of the user, because in the most of the configurations, there's no need to control all the ports in the same time.

METHODS

- **/ROOM:delete()**

  Format: :delete(<room_name>)

  Description: It deletes the given room.

  If a room is deleted while a controller connected to it, the controller will not disconnect, but will not be able to switch anymore.

  Note: The ROUTER room cannot be deleted.

- **/ROOM:create()**

  Format: :create(<room_name>)

Description: It creates a new room with the given name. The new room will be empty (ie. no layers and no ports will be added automatically). You have to add at least one media layer before you can setup the physical input and output list.

Note: The names are case sensitive, the maximum length is 32 character.

5.5.55. /ROOM/<name_of_room>/

Description: This node represents a room.

When a new room is created, it consists no media layer and it has no ports. You have to add new layers with the AddLayer function and set up the new layer's assign lists.

METHODS

- **/ROOM/<name_of_room>:addLayer()**

Format: :addLayer(<layer_id>)

Description: Every room may consist one or more media layer. When creating a new room, there will be no layer included. There must be added at least one layer before the physical assign list could be set up.

This method adds a new layer to the room. The new layer is identified by a number.

The available layers:

ID	layer
0	Video
1	Forward audio
2	Backward audio
3	Rs232
4	Infra
5	USB KVM

Example:

addLayer(1) - this will add the forward audio layer

Note: The available layers may depend on your license. If you have no license for specific layers, then the method will respond with error code 20. (E020:Layer not available)

- **/ROOM/<name_of_room>:deleteLayer()**

Format: :deleteLayer(<layer_id>)

Description: A layer can be removed from the room. The layer is identified by a number, see addLayer for the details.

Example:

deleteLayer(1) - this will remove the forward audio layer

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>.Description**

Format: .Description = <description>

Description: A freely editable description.

- **/ROOM/<name_of_room>.Name**

Format: .Name = <short_room_description>

Description: A freely editable name. This is independent from the room name, it is just for your convenience. The room name is usually a short abbreviation what can detailed here.

Note: When referring to the name of the room (for example when configuring the external controllers), always the name of the node should be used. The value of this property is not used anywhere.

READ-ONLY PROPERTIES

- **/ROOM/<name_of_room>.Size**

  Format: .Size = <number_of_inputs> ";" <number_of_outputs> ";"

  Description: This is the size of the room which is calculated based on input and output physical assign list.

  E.g. if the room has a 11 input and 20 output "011;020;"

  Note: If the sizes of the layers differ, then the greatest size will be returned.

5.5.56. /ROOM/<name_of_room>/EDID/

Description: EDID management related properties and methods.

METHODS

- **/ROOM/<name_of_room>/EDID:EmulateEDID()**

  Format: :EmulateEDID(<input_port_number> "," {"F" | "U" | "D"} <EDID_number>)

  Description: Emulates an EDID on an input port. The port is numbered inside the room, the location is an EDID index, include the 'U', 'D' or 'F' prefix.

  Example:

  EmulateEDID(1,F49) - Emulate the Factory 49 EDID on first input.

- **/ROOM/<name_of_room>/EDID:EmulateAll()**

  Format: :EmulateAll({"F" | "D" | "U"} <EDID_number>)

  Description: Emulates the given EDID on all inputs in the room.

  Example: EmulateAll(F1) - It emulates the Factory 1 EDID to all inputs in the room.

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>/EDID.EDIDLocations**

  Format: .EDIDLocations = *["F" | "D" | "U" <EDID_number> ";"]

  Description: This property contains as many EDID numbers as many video inputs are present in this room. The EDID numbers also contain the EDID types, such as 'F' (Factory), 'U' (User), 'D' (Dynamic).

  This property also can be written.

  Example:

  F49;U12;D21;F23;F49;

5.5.57. /ROOM/<name_of_room>/MONITOR/

Description: Settings and statuses related to the video monitor output

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>/MONITOR.SelectedPort**

  Format: .SelectedPort = { "N/A" | "EXT" | {{"I" | "O"} <logical_port_number>}

  Description: The actually monitored port. If there is no monitor board present, N/A value is reported back. If the actually monitored port is outside of the room, then the value is "EXT". Othervise the logical port number of the io port is sent.

  This property is writable: writing this property to the wanted value will change the monitor output.

  Example:

  SET /ROOM/ROUTER/MONITOR.SelectedPort=I12

The input 12 has been switched to the monitoring output.

READ-ONLY PROPERTIES

- **/ROOM/<name_of_room>/MONITOR.MobPresent**

  Format: .MobPresent = "true" | "false"

  Description: Is there a monitoring board present in the system? If not, the monitoring function couldnt be used.

5.5.58. /ROOM/<name_of_room>/NAMES/

Description: The user is able to attach names to the ports of the matrix. Each room has this node, but the names are global for the system, so please note, that their modification can affect all rooms which contains the port.

5.5.59. /ROOM/<name_of_room>/NAMES/<name_of_layer>/

Description: The node name represents a layer. The available layers depends on the room settings.

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>/NAMES/<name_of_layer>.I<logical_port_number>**

  Format: .I = <bitmap_id> ";" <name_of_port>

  Description: This property holds the name of an input in the selected layer. The name consists two parts (separated by semicolon): first is the ID of the bitmap which is used in the control software, the second one is the name itself.

  Example:

  /ROOM/test/NAMES/VIDEO.I3=2;Teacher laptop

  Note: Please note that I/O names are global in the matrix. If you modify an I/O name, then the given port will change the name in all room, where it is present. Therefore don't give write rights to this node unless this is you want.

- **/ROOM/<name_of_room>/NAMES/<name_of_layer>.O<logical_port_number>**

  Format: .O = <bitmap_id> ";" <name_of_port>

  Description: This property holds the name of an output in the selected layer. The name consists two parts (separated by semicolon): first is the ID of the bitmap which is used in the control software, the second one is the name itself.

  Note: Please note that I/O names are global in the matrix. If you modify an I/O name, then the given port will change the name in all room, where it is present. Therefore don't give write rights to this node unless this is you want.

- **/ROOM/<name_of_room>/NAMES/<name_of_layer>.V<logical_port_number>**

  Format: .V = <bitmap_id> ";" <name_of_port>

  Description: This property holds the name of a virtual control port in the selected layer. The name consists two parts (separated by semicolon): first is the ID of the bitmap which is used in the control software, the second one is the name itself.

5.5.60. /ROOM/<name_of_room>/PORTS/

METHODS

- **/ROOM/<name_of_room>/PORTS:FactoryDefaults()**

  Description: Resets all ports of all layers in the room to default settings.

5.5.61. /ROOM/<name_of_room>/PORTS/<name_of_layer>/

METHODS

- **/ROOM/<name_of_room>/PORTS/<name_of_layer>:FactoryDefaults()**

Description: Resets all ports of on the layer in the room to default settings.

5.5.62. /ROOM/<name_of_room>/PORTS/<name_of_layer>/I<logical_port_number>/

Description: This node covers the actual status information and settings of the given input port. The exact methods and properties are depending on the input board type. Please see the next section for the detailed description of available input boards.

5.5.63. /ROOM/<name_of_room>/PORTS/<name_of_layer>/O<logical_port_number>/

Description: This node covers the actual status information and settings of the given output port. The exact methods and properties are depending on the output board type. Please see the next section for the detailed description of available output boards.

5.5.64. /ROOM/<name_of_room>/PORTS/<name_of_layer>/V<logical_port_number>/

Description: This node covers the actual status information and settings of the given virtual control port. The exact methods and properties are depending on the actual layer. Virtual control ports are available for INFRA and RS232 layers at this moment. The detailed descriptions of the available properties and methods can be found under the /MANAGEMENT/VCP nodes.

See also: /MANAGEMENT/VCP

5.5.65. /ROOM/<name_of_room>/PRESET/

Description: This node holds the presets within the given room.

METHODS

- **/ROOM/<name_of_room>/PRESET:create()**

Format: :create(<name_of_preset> "," 1*[<layer_name> ","])

Description: It creates a new empty preset, with the media layers defined by the parameter list. The available layer names:

- VIDEO
- FWAUDIO
- RETAUDIO
- RS232
- INFRA
- KVM

Note: In the parameters list, the first parameter is the name of the preset, then the list of the layers are separated by a comma [,].

The available layer names can be limited by the installed licenses.

- **/ROOM/<name_of_room>/PRESET:delete()**

Format: :delete(<name_of_preset>)

Description: It deletes a preset.

Note: Preset names are case sensitive.

- **/ROOM/<name_of_room>/PRESET:copy()**

Format: :copy(<name_of_source_preset> "," <name_of_destination_preset>)

Description: It copies (clones) the source preset into the destination preset.

Note: Preset names are case sensitive.

The destination preset cannot exist.

5.5.66. /ROOM/<name_of_room>/PRESET/<name_of_preset>/

Description: A preset in the given room. A preset may contain one or more layer from the room. If a layer is present in the preset, then it stores every connection on that layer and it contains also the source and destination mute states. When a preset is loaded, then every earlier connection will be ceased in the room and the stored connections will be created, while the source and destination mute states will be updated.

METHODS

- **/ROOM/<name_of_room>/PRESET/<name_of_preset>:save()**

Description: It saves the current crosspoint state into the preset.

Note: It saves all of the layers defined by the preset.

- **/ROOM/<name_of_room>/PRESET/<name_of_preset>:load()**

Description: Loads the preset.

Note: It loads all of the layers defined by the preset.

- **/ROOM/<name_of_room>/PRESET/<name_of_preset>:addLayer()**

Format: :addLayer(<layer_id>)

Description: This method adds a new layer to the preset.

It automatically saves the current crosspoint state to the properties.

The text argument is the number of the layers:

layer_id	layer name
0	VIDEO
1	FORWARD AUDIO
2	RETURN AUDIO
3	RS232
4	INFRA
5	KVM

- **/ROOM/<name_of_room>/PRESET/<name_of_preset>:deleteLayer()**

Format: :deleteLayer(<layer_id>)

Description: This method removes the specified layer from the preset.

The text argument is the number of the layers:

The layer_id codes are described at the addLayer() method

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>/PRESET/<name_of_preset>.<layer_name>Source**

Format: .Source = <source_number>*{{"u"|"m"} ";"}

Description: This property defines the mute state of the sources by a semicolon separated list. The items can be the next two character:

Character	Meaning
u	unmute
m	mute

In the Presets either unmute or mute must be specified (it is not possible that one of the ports remains unchanged).

Note: All of the ports within the layer must be listed. (The order of the ports: input first, output first, only inputs, only outputs are layer dependent).

- **/ROOM/<name_of_room>/PRESET/<name_of_preset>.<layer_name>Destination**

Format: .Destination = <number_of_destinations>*{{"m" | "u"} {{{"I" | "O" | "V"} <source_number>} | "0" } ";" }

Description: It defines the connection and the mute state of the destinations. This property is a semicolon separated list, each item represents the destinations of the room in ascending order. The actual destinations are depending on the layer type, e.g. in a video layer the destinations can be only the outputs (O1, O2, O3, ... On) while an RS232 or INFRA layer can have destination on input ports too. In that case outputs preceding the inputs (O1, O2 ... On, I1, I2, .. In). If virtual control ports are also added to the room, they will be placed at the end of the list.

The mute/unmute state has to be defined by the next characters:

Character	Meaning
u	unmute
m	mute

Either unmute or mute must be specified (it is not possible that one of the ports remains unchanged). After that the connected source must be specified. On the layers where multipoint-point connections are allowed, the source can be also a list itself separated by colons. The source can be specified by the next ways:

Character	Meaning
I<logical_port_id>	Input port (physically on an input board)
O<logical_port_id>	Output port (physically on an output board)
V<logical_port_id>	Virtual port (Virtual control port)
0 (zero)	Not connected

Example:

/ROOM/boarding/PRESETS/conferencing.VideoDestination=uI1;uI12;uI13;mI20

In the boarding room the conferencing presets stores for the video layer the next connections: I1 is connected to O1 (unmuted), I12 is connected to O2 (unmuted), I13 is connected to O3 (unmuted) and I20 is connected to the muted O4 port.

Example:

/ROOM/boarding/PRESETS/conferencing.UsbKVMDestination=uI1;uI12,I4;

In the boarding room the conferencing presets stores for the USB KVM layer the next connections: I1 is connected to O1 (unmuted), I12 and I4 is connected to the unmuted O2 port.

5.5.67. /ROOM/<name_of_room>/SALVO/

Description: This node holds the salvos within the given room.

METHODS

- **/ROOM/<name_of_room>/SALVO:create()**

Format: :create(<name_of_salvo> "," 1*[<layer_name> ","])

Description: It creates a new empty salvo, with the media layers defined by the parameter list.

Note: In the parameters list, the first parameter is the name of the preset, then the list of the layers separated by a comma.

Salvo names are case sensitive.

- **/ROOM/<name_of_room>/SALVO:delete()**

  Format: :delete(<salvo_name>)

  Description: It deletes the salvo with the given name.

  Note: Salvo names are case sensitive.

- **/ROOM/<name_of_room>/SALVO:copy()**

  Format: :copy(<string>,<string>)

  Description: It copies (clones) the source salvo (first parameter) into the destination salvo (second parameter).

  Example:

  :copy(oldOne,newOne)

  Note: Salvo names are case sensitive.

  The destination salvo must not exist.

5.5.68. /ROOM/<name_of_room>/SALVO/<name_of_salvo>/

Description: A salvo within the given room.

METHODS

- **/ROOM/<name_of_room>/SALVO/<name_of_salvo>:load()**

  Description: Loads the salvo.

  Note: It loads all of the layers defined by the salvo.

- **/ROOM/<name_of_room>/SALVO/<name_of_salvo>:deleteLayer()**

  Description: This method removes the specified layer from the salvo.

  See also: /ROOM/<name_of_room>/PRESET/<name_of_preset>:addLayer()

- **/ROOM/<name_of_room>/SALVO/<name_of_salvo>:addLayer()**

  Format: :addLayer(<layer_id>)

  Description: This method adds a new blank layer to the salvo (creates the [LayerName]Destination and [LayerName]Source properties).

  See also: /ROOM/<name_of_room>/PRESET/<name_of_preset>:addLayer()

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>/SALVO/<name_of_salvo>.<layer_name>Destination**

  Format: .Destination = <number_of_destinations>*{ [{"m" | "u"}] [{ { {"I" | "O" | "V"} <source_number>} | "0" }] ";" }

  Description: It defines the connection and the mute state of the destinations. This property is a semicolon separated list, each item represents the destinations of the room in ascending order. The actual destinations are depending on the layer type, e.g. in a video layer the destinations can be only the outputs (O1, O2, O3, ... On) while a RS232 or INFRA layer can have destination on input ports too. In that case outputs preceding the inputs (O1, O2 ... On, I1, I2, .. In). If virtual control ports are also added to the room, they will be placed at the end of the list.

  The mute/unmute state can be optionally defined by the next characters for each destinations:

Character	Meaning
u	unmute
m	mute

The connected source also can be specified. On the layers where multipoint-point connections are allowed, the source can be also a list itself separated by colons. The source can be specified by the next ways:

Character	Meaning
I<logical_port_id>	Input port (physically on an input board)
O<logical_port_id>	Output port (physically on an output board)
V<logical_port_id>	Virtual port (Virtual control port)
0 (zero)	Not connected

If the connections of a destination must be left unchanged, the item can be left empty.

Example:

/ROOM/boarding/PRESETS/videoconference.VideoDestination=I1;;uI13;m

In the boarding room the videoconference salvo switches I1 to O1 (while the mute/unmute state of O1 has been left unchanged), the connection of the O2 output port is preserved, the I13 input is switched to O3 and O3 is unmuted and at last the port O4 is muted. The boarding room has exactly 4 destinations on the video layer.

- **/ROOM/<name_of_room>/SALVO/<name_of_salvo>.<layer_name>Source**

Format: .Source = <number_of_sources>*{["u" | "m"] ";"}

Description: This property defines the mute state of the sources by a semicolon separated list. The items can be the next two character:

Character	Meaning
u	unmute
m	mute
(empty)	leaving unchanged

Example:

/ROOM/boarding/SALVO/init.VideoSource=u;;;;m;;;;m;;;;;;

The init salvo unmutes the first video source and mutes the 5th and 10th video source in the boarding room.

Note: The number of the list elements must be the same as the number of the sources on the specified layer.

All of the ports within the layer must be listed, but it is possible to leave some of them blank (see example above).

5.5.69. /ROOM/<name_of_room>/SETTINGS/

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>/SETTINGS.InputAssignList**

Note: This property is used for debug purposes. It's not recommended to modify the property's value.

- **/ROOM/<name_of_room>/SETTINGS.OutputAssignList**

Note: This property is used for debug purposes. It's not recommended to modify the property's value.

- **/ROOM/<name_of_room>/SETTINGS.VCPAssignList**

Note: This property is used for debug purposes. It's not recommended to modify the property's value.

5.5.70. /ROOM/<name_of_room>/SETTINGS/<name_of_layer>/

READ&WRITE PROPERTIES

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.InputAssignList**

  Format: .InputAssignList = *[["I"] <input_port_number> ";"]

  Description: This property defines which physical input ports are parts of the room on this layer. This property holds a semicolon separated input port list.

  Example: I5;I12;I120;I159; -- the last semicolon is optional

  The order of the ports is important - they determine the logical port numbering. In the above example, physical input 5 will be the first input in the room, 12 the second one and so on.

  This property always refer to the input boards, regardless of is it a source on the given layer or is it a destination. For example, backward audio layer has the destinations on the input boards, which should defined in this property.

  It is also possible to skip ports by writing zeroes. In that case the given logical port will be not mapped to any physical port, but it is working as purely virtual. It is also even possible to define a room without any physical port.

  Example: I5;0;0;I159;

  In that case the logical second and third inputs have no physical correspondence.

  With RS232 and infra layers it is also possible using subports: a connected remote extender (MODEX) may have more serial or IR ports which have to be distinguished. Let assume that a MODEX is connected to the *n*.th input and it has four serial ports. In that case these serial ports can be referred as I*n*.1, I*n*.2, I*n*.3, I*n*.4. If there is no explicit subport definition, then it refers to the first subport, so I*n* is equal to I*n*.1

  Example: I5;I12.3;I120;I159;

  In that case the second logical input uses the third remote serial port, while others use the first ones.

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.OutputAssignList**

  Format: .OutputAssignList = *[["O"] <output_port_number> ";"]

  Description: This property defines which physical output ports are parts of the room on this layer. This property holds a semicolon separated output port list.

  Example: O5;O12;O120;O159; -- the last semicolon is optional

  For further detailed explanation see the InputAssignList description above.

  Everything is written here can be applied to the OutputAssignList too.

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.VcpAssignList**

  Format: .VcpAssignList = *[["V"] <vcp_port_number> ";"]

  Description: This property defines which physical virtual control ports are parts of the room on this layer. This property holds a semicolon separated virtual control port list. This property is used to rs232 and infra layer only, other layers must have empty VcpAssignList.

  Example: V2;V4;V6;V9; -- the last semicolon is optional

  The order of the ports determines the logical port numbering. In the above example, VCP 2 will be the first virtual control port in the room, 4 the second one and so on.

  It is also possible to skip ports by writing zeroes. In that case the given logical port will be not mapped to any physical port, but it is working as purely virtual.

  Example: V5;0;0;V9;

  VCP ports have not subports.

READ-ONLY PROPERTIES

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.MaxDstToSrc**

  Format: .MaxDstToSrc = <max_dest_number> | "oo"

  Description: The maximum number of destinations that can be connected to one source at same time in this layer.

  "oo" represents the infinity value

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.MaxSrcToDst**

  Format: .MaxSrcToDst = <max_source_number> | "oo"

  Description: The maximum number of sources that can be connected to one destination at same time in this layer.

  "oo" represents the infinity value

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.Subports**

  Format: .Subports = <subport_number>

  Description: The maximum available subports on this layer.

  Note: This value is 16 for RS232 and infra ports, while it is zero for other layers as they do not have subports.

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.VcpPorts**

  Format: .VcpPorts = <vcp_number>

  Description: The maximum number of available virtual control ports on this layer.

  Note: This value is non-zero only for RS232 and infra layer as other layers don't support virtual control ports.

- **/ROOM/<name_of_room>/SETTINGS/<name_of_layer>.Layout**

  Format: .Layout = 4*{<0 | 1> ";"}

  Description: This property describes whether inputs and/or outputs can be used as a source and/or destination on this layer. This information is represented by four binary value separated by semicolon. The meaning of these binary values:

Position	Meaning
1	It is one if input ports can operate as sources, zero otherwise
2	It is one if output ports can operate as sources, zero otherwise
3	It is one if input ports can operate as destinations, zero otherwise
4	It is one if output ports can operate as destinations, zero otherwise

Examples:

Layer	Value
Video	1;0;0;1;
Backward audio	0;1;1;0;
Rs232	1;1;1;1;

5.5.71. /ROOM/<name_of_room>/XP/

Description: This node represents the crosspoint of the room. Each layer in the room will have a subnode here (with the name of the layer) where the crosspoint state can be read back and/or can be changed.

5.5.72. /ROOM/<name_of_room>/XP/<name_of_layer>/

METHODS

- **/ROOM/<name_of_room>/XP/<name_of_layer>:connect()**

Format: :connect(1*[1*[["-"] {"I" | "O" | "V"} <logical_port_number> ","] {":" | "x"} 1*[["-"] {"I" | "O" | "V"} <logical_port_number> ","]])

Description: Add or remove multiple connections in the room. There are two request types: the disconnect, which can be used to remove connections, and the connect that can be used to add connections.

The disconnect request is defined by a comma-separated list of sources followed by an "x" character and a comma separated list of destinations. The listed sources will be disconnected from the listed destinations.

Example:

CALL /ROOM/ROUTER/XP/VIDEO:connect(I1,I2,I3xO1,O2,O3)

Every connection will be ceased between I1,I2,I3 input ports and O1,O2,O3 output ports. The connection request is defined by a comma-separated list of sources followed by a colon and a comma-separated list of destinations. The listed sources will be connected to the listed destinations. If a source or destination reached the maximum number of available connections and hence the new connections cannot be added, the current connections remain unchanged. Optionally a dash ("-") can be placed in front of the listed sources and destinations. In this case all of the current connections will be removed before the requested connections are added.

Example:

CALL /ROOM/ROUTER/XP/VIDEO:connect(I1:-O1,-O2,O3)

The I1 input port is switched to the O1, O2, O3 output ports in the ROUTER room. The old connections from the O1 and O2 ports will be removed. If the O3 had a connection before, then connecting I1 to O3 will be failed hence the multipoint-point style connections are not supported on the video layer.

Example:

CALL /ROOM/ROUTER/XP/VIDEO:connect(-I1:-O1)

I1 is connected to the O1 port now. Every earlier connections will be removed from both I1 and O1. (if I1 has been connected to other output ports before, then they are disconnected now) The method can take multiple semicolon separated requests. Requests are evaluated from left to right. Changes are performed on a temporary buffer first. Subsequent requests may overwrite the effects of preceding ones. After processing the last request, changes are performed in an atomic fashion on the physical crosspoint.

Example:

CALL /ROOM/ROUTER/XP/FWAUDIO:connect(I1,I2xO1,O2;I3xO5;I1:-O7;-I8:-O6,O9)

This connect command has four steps. First step: I1,I2xO1,O2 will disconnect I1 and I2 from both O1 and O2. Other existing connections of the I1, I2, O1, O2 ports will not be modified. Second step: I3xO5 will disconnect I3 from O5. Other existing connections of the I3 and O5 ports will not be modified. Third step: I1:-O7 will first disconnect all sources from the O7 port and then connect only the I1. Other existing connections of the I1 ports will not be modified. Fourth step: -I8:-O6,O9 will first disconnect all destinations from the I8, then disconnect all sources from the O6 and at the end the I8 will be connected to the O6 and O9 ports. Other existing connections of the O9 ports will not be modified.

It is important to understand that the created connections are unidirectional. This makes sense for video and audio, but needs extra attention for example if using RS232 layer: if a bidirectional connection is required, both direction must be connected:

CALL /ROOM/ROUTER/XP/RS232:connect(I1:O1;O1:I1)

If a layer has virtual control ports, then they can be switched in the same manner.

Example:

CALL /ROOM/ROUTER/XP/RS232:connect(-V1:-O1;O1:V1)

Virtual control port 1 is connected to O1. The connection is bidirectional (sending/receiving data is also possible). Every other connections were removed from V1 and O1.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:muteSource()**

  Format: :muteSource(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method mutes one or more source ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already muted ports will not be modified. Locked ports cannot be muted.

  Example:

  CALL /ROOM/ROUTER/XP/RS232:muteSource(I1;I5;O6)

  It will try to mute the I1, I5 and O6 source ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:muteDestination()**

  Format: :muteDestination(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method mutes one or more destination ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already muted ports will not be modified. Locked ports cannot be muted.

  Example:

  CALL /ROOM/ROUTER/XP/VIDEO:muteDestination(O1;O5;O6)

  It will try to mute the O1, O5, and O6 ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:lockSource()**

  Format: :lockSource(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method locks one or more source ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already locked ports will not be modified.

  Example:

  CALL /ROOM/ROUTER/XP/VIDEO:lockSource(I1;I5;I6)

  It will try to lock the I1, I5 and I6 ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:lockDestination()**

  Format: :lockDestination(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method locks one or more destination ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already locked ports will not be modified.

  Example:

  CALL /ROOM/ROUTER/XP/VIDEO:lockDestination(O1;O5;O6)

  It will try to lock the O1, O5 and O6 ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:switch()**

  Format: :switch(1*[1*[{"O" | "I" | "V"} <logical_port_number> ","] ":" {"O" | "I" | "V"} <logical_port_number> ";"])

  Description: This method is similar to the connect method. It connects one or more sources to one destination. All previously connected sources get disconnected form the destination.

  To disconnect the destination from all its sources set the source to "0" (zero).

The source ports have to be separated by a comma, then a colon follows and at the end there is only one destination.

More than switch command can be listed in blocks. The blocks are executed from left to right order. The blocks are separated by semicolon.

If any of the ports within one block is locked, hence the switch cannot be done, the execution of the block is terminated and nothing will be changed. The termination doesn't interrupt, so next blocks will be processed.

Example:

CALL /ROOM/test/XP/VIDEO:switch(I1:O1;I2:O2;)

This will first disconnect all of the current connections from the O1 and then connect I1 to the O1. Then disconnect everything from O2 and connect it to I2. If any of the above blocks fails, the other one will be executed nevertheless.

Example:

CALL /ROOM/test/XP/RS232:switch(I1,I2,I3:O1;I1,I4:O3)

I1,I2,I3:O1 will first disconnect all of the current connections from the O1 and then connect I1, I2, I3 to the O1. Then I1,I1:O3 will first disconnect all of the current connections form the O3 and then connect I1 and I4 to the O3.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:switchMulti()**

  Format: :switchMulti(1*[*[{"O" | "I" | "V"} <port_number> ","] ";"])

  Description: Switch multiple sources to multiple destinations in the room. The method takes a comma separated list of sources for each destination in the room. The listed sources will be connected to the specific destination. All current connections from the specified destination will be first disconnected. If the list element is empty (see the example), the connections of that destination will not be changed. The first source list is assigned to the first destination, the second one is to the second, and so on. Source lists are semicolon separated.

  If any of the destination or source is locked within a block (semicolon separated list element) that block is not executed and none of the ports within that block is modified.

  Example:

  CALL /ROOM/test/XP/USBKVM=I1,I3;I5;;0;;I6;

  The order of the execution: Disconnect all connections from the 1st destination and then connect I1 and I3. Disconnect all connections from the 2nd destination and then connect the I5. No modification on the 3rd destination. Disconnect all connections from the 4th destination. No modification on the 5th destination. Disconnect all connections from the 6th destination and then connect the I6. No modification on the remaining destinations if any.

  Note: This method is efficient in reconfiguring all the connections in the room because it eliminates the overhead of explicitly specifying the destinations.

  This method can process the value of DestinationConnectionStatus property directly.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:switchAll()**

  Format: :switchAll({{"I" | "O" | "V"} <logical_port_number>} | "0")

  Description: This method disconnect all of the current connections from all destinations in the room and connect only one source to all of them or leaves them unconnected.

  If any of the destinations are locked only that specific one will not be modified.

  If the specified source is locked the execution is terminated without any changes.

The parameter can be any of the followings:

Source	Description
I<logical_port_number>	An input port (physically on an input board)
O<logical_port_number>	An output port (physically on an output board)
V<logical_port_number>	A virtual control port
0	Disconnect

Example:

CALL /ROOM/ROUTER/XP/VIDEO:switchAll(I1)

If the input port 1 is unlocked, then every unlocked output port will be switched to the input port 1 in the ROUTER room.

Example:

CALL /ROOM/ROUTER/XP/VIDEO:switchAll(0)

Every unlocked output port will be disconnected.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:unmuteSource()**

  Format: :unmuteSource(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method unmutes one or more source ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already unmuted ports will not be modified. Locked ports cannot be unmuted.

  Example:

  CALL /ROOM/ROUTER/XP/RS232:unmuteSource(I1;I5;O6)

  It will try to unmute the I1, I5 and O6 source ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:unmuteDestination()**

  Format: :unmuteDestination(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method unmutes one or more destination ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already unmuted ports will not be modified. Locked ports cannot be unmuted.

  Example:

  CALL /ROOM/ROUTER/XP/VIDEO:unmuteDestination(O1;O5;O6)

  It will try to unmute the O1, O5, and O6 ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:unlockSource()**

  Format: :unlockSource(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method unlocks one or more source ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already unlocked ports will not be modified.

  Example:

  CALL /ROOM/ROUTER/XP/VIDEO:unlockSource(I1;I5;I6)

  It will try to unlock the I1, I5 and I6 ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>:unlockDestination()**

  Format: :unlockDestination(*[{"O" | "I" | "V"} <port_number> ";"])

  Description: This method unlocks one or more destination ports. The number of the elements can be vary and the elements are separated by semicolons.

  Already unlocked ports will not be modified.

Example:

CALL /ROOM/ROUTER/XP/VIDEO:unlockDestination(O1;O5;O6)

It will try to unlock the O1, O5 and O6 ports.

READ-ONLY PROPERTIES

- **/ROOM/<name_of_room>/XP/<name_of_layer>.DestinationConnectionStatus**

Format: .DestinationConnectionStatus = <number_of_destinations>*{{{{"I" | "O" | "V"} <logical_port_number>} | "E" | 0} ";"}

Description: This property represents the actual state of the crosspoint for the room. This property is a list, containing the same number of elements as the number of the destinations in the room. Each element represents a destination and specify to which source it is connected. The elements are separated by semicolon.

However on the video layer only output ports can be destinations, therefore the number of items in this property matches with the number of output ports in the room, other layers may have a more complex structure. For example, on an infra or RS232 layer both inputs and outputs can serve as destination, and in addition some virtual control port also may be present. The general rule is that outputs (if present) are preceding the inputs, and virtual control ports are at the end of the list. If our imaginary room has 3 outputs, 2 inputs and 3 VCPs, then the DestinationConnectionStatus will contain 8 items respectively to O1,O2,O3,I1,I2,V1,V2,V3 destination ports.

On those special layers where the multipoint-multipoint connections are allowed, one element can be a list of the sources connected to the destination. In this case the sources are separated by a colon.

The sources can be defined as the followings:

Source	Description
I<logical_port_number>	An input port (physically on an input board)
O<logical_port_number>	An output port (physically on an output board)
V<logical_port_number>	A virtual control port
0	There is no connection to this destination
E	This connection is coming from outside of the room

Example:

/ROOM/test/XP/VIDEO.DestinationConnectionStatus=I12;I5;I6;

There are three destinations defined in the room. The first is connected to the Input port 12, the second to the Input port 5 and the third to the Input port 6.

Important: The port numbers here are logical port numbers inside the room and nothing to do with the actual physical port number. The only thing we know is that all source ports are physically on input boards.

Example:

/ROOM/test/XP/INFRA.DestinationConnectionStatus=I1,I5,I6;I8;I9,I4;0;0;0;0;0;0;0;0;0;E;

There are 16 destinations in this room on the infra layer. The first is connected to the input port 1, 5 and 6 at same time. The second is connected only to the Input port 8, while the third one to the Input port 9 and 4. The other destinations are unconnected, except the last one, which has an external connections (outside from the room, we do not know the physical port).

Example:

Let assume that our test room has 2 inputs, 2 outputs and 2 VCP ports on the RS232 layer. As this layer is bidirectional, we have six destination now.

/ROOM/test/XP/RS232.DestinationConnectionStatus=I1;0;O1;0;I1;I2;

The first two item represents the output ports, so I1 is connected to the O1, while O2 has no incoming connection now. The second two item represents the inputs, therefore O1 is connected to the I1, while I2 has no incoming connections. The last two items are for virtual control ports: I1 is connected to V1, while I2 is connected to V2 now. To summarize: I1 and O1 has a bidirectional connection, while I1 and I2 are monitored on the V1 and V2 virtual control ports.

- **/ROOM/<name_of_room>/XP/<name_of_layer>.SourcePortStatus**

  Format: .SourcePortStatus = <number_of_sources>*{{"T" | "L" | "U" | "M"} <hex_number> ";" }

  Description: This property represents the port statuses for the source ports in a room. This property is a list, containing the same number of elements as the number of the sources on the layer in the room (this can be different for different layer, because of the bidirectional ports). The list is separated by semicolon.

  The first character of an element is one of the followings:

Character	Meaning
M	The port is muted.
L	The port is locked.
U	Both muted and locked.
T	None.

After the first character there is a hexadecimal number, representing 16 bits. The meaning of the value is different for the different layers, but the representation is the same for all of them: the endianness is bigendian and the leading zeros can be omitted. In the 16 bits there are bit pairs defined with the following meanings:

Bit pair	Meaning
00	Unknown (the physical layer is not capable measuring the property)
01	RFU
10	False
11	True

The most significant bit (MSB) in the 16 bits indicates if there is a hardware failure detected:

MSB value	Meaning
0	Failure
1	the board is working properly

The meaning of the bit pairs for the different layers (MSB first):

Bit pair	Meaning on the video layer
15 (MSB)	Hardware failure
14:13	Reserved, 00 as default
12:11	Reserved, 00 as default
11:10	Reserved, 00 as default
9:8	Reserved, 00 as default
7:6	Embedded Audio in the Video Stream
5:4	The Video content is HDCP encrypted
3:2	Video Signal Present
1:0	The port is connected

Besides the four digit hexadecimal number the port status can be a dash (-). This means that there is no physical board installed for that port.

Example:

/ROOM/test/XP/VIDEO.SourcePortStatus=T2F;L2A;T-;

There are three sources in the test room, however there is no operational input board on the third port. The first one is unmuted and unlocked, the second one is unmuted and locked. The first one has the port status 0x2F (the leading zeros can be omitted: 002F is written as 2F), while the second one has 2A. As 2F is equal to 00 10 11 11 in binary form, while 2A is equal to 00 01 10 10, the decoded meaning is summarized in the table below:

Bit pair	First port (002F)	Second port (002A)
15	0 - No hw failure	0 - No hw failure
7:6	00 - No info about audio	00 - No info about audio
5:4	10 - No HDCP	10 - No HDCP
3:2	11 - Signal present	10 - No signal
1:0	11 - Connected	10 - Not connected

- **/ROOM/<name_of_room>/XP/<name_of_layer>.DestinationPortStatus**

  Format: .DestinationPortStatus = <number_of_destinations>*{{"T" | "L" | "U" | "M"} <hex_number> ";" }

  Description: This property represents the port status for the destination ports in a room. This property is a list, containing the same number of elements as the number of the destinations on the layer in the room (this can be different for different layer, because of the bidirectional ports). The list is separated by semicolon.

  The value of this property can be interpreted in the same way as the "SourcePortStatus". See a detailed description and example there.

5.5.73. /ROOM/ROUTER/

Description: This room represents the whole matrix. It cannot be deleted or removed. The assign lists cannot be changed in any kind.

This room consists the all available layer and all available ports. The room ports are analogous to the physical port numbers. The ROUTER room doesn't contain any VCP (Virtual Control Port), if VCPs are needed then the user must create its own room.

See also: /ROOM/<name_of_room>

5.6. 25G-8HDMI1-IB and 25G-8DVID1-IB

5.6.1. Video layer

Description: This node contains the video stream parameters of the node.

READ&WRITE PROPERTIES

- **I<logical_port_number>.InputHdcpEnable**

  Format: .InputHdcpEnable = "true" | "false"

  Description: Enables/Disables the input port's HDCP capability. If it's set to "false" the source will detect that the port does not support HDCP, thus will not transmit HDCP encrypted video.

READ-ONLY PROPERTIES

- **I<logical_port_number>.SignalPresent**

  Format: .SignalPresent = 0 | 1

  Description: Contains "1" if there is a valid video signal present on the port.

- **I<logical_port_number>.SignalType**

  Format: .SignalType = 0 | 1

  Description: Contains the type of the transmitted video signal.

Value	Meaning
0	The video signal is HDMI.
1	The video signal is DVI.

- **I<logical_port_number>.HdcpActive**

  Format: .HdcpActive = 0 | 1

  Description: Shows the HDCP encryption state of the video signal.

Value	Meaning
0	The video signal is not HDCP encrypted.
1	The video signal is HDCP encrypted.

- **I<logical_port_number>.Resolution**

  Format: .Resolution = <horizontal_resolution> "x" <vertical_resolution>

  Description: Contains the active resolution of the transmitted video signal.

  Example:

  I1.Resolution=1280x720

  This means that the active region of the video stream contains 1280 columns and 720 rows.

- **I<logical_port_number>.TotalSize**

  Format: .TotalSize = <horizontal_resolution> "x" <vertical_resolution>

  Description: Contains the total resolution (active region and blanking region) of the transmitted video signal.

  Example:

  I1.Resolution=1980x750

  This means that the video stream contains 1980 columns and 750 rows. These values include the size of the blanking regions.

- **I<logical_port_number>.ColorDepth**

  Format: .ColorDepth = 0 | 1 | 2 | 3

  Description: Color depth of the transmitted video signal

Value	Meaning
0	Color depth is 24 bits/pixel
1	Color depth is 30 bits/pixel
2	Color depth is 36 bits/pixel
3	Color depth is 48 bits/pixel

- **I<logical_port_number>.Scan**

  Format: .Scan = 0 | 1

  Description: Indicates that the scan of the video signal is progressive or interlaced.

Value	Meaning
0	Progressive
1	Interlaced

- **/ I<logical_port_number>.Power5vIn**

  Format: .Power5vIn = "true" | "false"

  Description: "True", if 5V voltage is sensed on the input. This means that the source is connected and powered.

5.7. 25G-8HDMI1-OB and 25G-8DVID1-OB

5.7.1. Video layer

Description: This node contains the video stream parameters of the node.

READ&WRITE PROPERTIES

- **O<logical_port_number>.OutputMode**

Format: .OutputMode = "DVI" | "HDMI" | "AUTO"

Description: Sets the output mode of the port.

Value	Meaning
"DVI"	The output is DVI format (no HDMI specific frames such as embedded audio, blanking area is empty).
"HDMI"	The output is HDMI.
"AUTO"	The output is: ▪ DVI for DVI receivers, ▪ HDMI when embedded audio is transmitted, ▪ Otherwise matches the input format.

- **O<logical_port_number>.EnableAudioEmbedding**

Format: .EnableAudioEmbedding = "true" | "false"

Description: Enables or disables embedding input S/PDIF audio into the HDMI video stream. Input audio source is jumper-selectable.

READ-ONLY PROPERTIES

- **O<logical_port_number>.SignalPresent**

Format: .SignalPresent = 0 | 1

Description: Contains "1" if there is a valid video signal present on the port.

- **O<logical_port_number>.SignalType**

Format: .SignalType = 0 | 1

Description: Contains the type of the transmitted video signal.

Value	Meaning
0	The video signal is HDMI.
1	The video signal is DVI.

- **O<logical_port_number>.HdcpActive**

Format: .HdcpActive = 0 | 1

Description: Shows the HDCP encryption state of the video signal.

Value	Meaning
0	The video signal is not HDCP encrypted.
1	The video signal is HDCP encrypted.

- **O<logical_port_number>.Resolution**

Format: .Resolution = <horizontal_resolution> "x" <vertical_resolution>

Description: Contains the active resolution of the transmitted video signal.

Example:

O1.Resolution=1280x720

This means that the active region of the video stream contains 1280 columns and 720 rows.

- **O<logical_port_number>.TotalSize**

Format: .TotalSize = <horizontal_resolution> "x" <vertical_resolution>

Description: Contains the total frame (active and blanking region) of the transmitted signal.

Example:

O1.Resolution=1980x750

This means that the video stream contains 1980 columns and 750 rows. These values include the size of the blanking regions.

- **O<logical_port_number>.ColorDepth**

  Format: .ColorDepth = 0 | 1 | 2 | 3

  Description: Color depth of the transmitted video signal

Value	Meaning
0	Color depth is 24 bits/pixel
1	Color depth is 30 bits/pixel
2	Color depth is 36 bits/pixel
3	Color depth is 48 bits/pixel

- **O<logical_port_number>.Scan**

  Format: .Scan = 0 | 1

  Description: Indicates that the scan of the video signal is progressive or interlaced.

Value	Meaning
0	Progressive
1	Interlaced

- **O<logical_port_number>.HotPlugDetect**

  Format: .HotPlugDetect = "true" | "false"

  Description: "True" indicates that there is a HDMI/DVI cable between the port's connector and a sink device.

- **O<logical_port_number>.ReceiverSense**

  Format: .ReceiverSense = "true" | "false"

  Description: "True" indicates that the TMDS lines are terminated by the sink device (i.e. the sink device is turned on).

5.8. 25G-8HDMI2-A1-IB

5.8.1. Main node

Path: /ROOM/<name_of_room>/PORTS/VIDEO/I<port_number>

Example: /ROOM/ROUTER/PORTS/VIDEO/I65

The node contains the following properties and method:

- **ProductName**

  The name of the I/O board which has the current port.

- **ProductSerialNumber**

  The serial number of the I/O board which has the current port.

- **ProductPartNumber**

  The part number of the I/O board which has the current port.

- **FirmwareVersion**

  The firmware version of the I/O board which has the current port.

- **getXml**

  <To be defined>

5.8.2. Parameters node

Path: /ROOM/<name_of_room>/PORTS/VIDEO/I<port_number>/PARAMETERS

Example: /ROOM/ROUTER/PORTS/VIDEO/I65/PARAMETERS

This node contains the video stream parameters of the given input port. The node contains the following properties and methods:

- **Connected**

  Cable present status. "0" means no cable is connected, "1" means cable is connected, "F" means the cable present status is unknown.

- **SignalPresent**

  The property is "true" if there is a valid signal on the video port, otherwise "false".

- **SignalType**

Value	Meaning
0	DVI
1	HDMI
2	DP-SDI

- **SignalSubType – BartaG (Dual link?)**

  The property has meaning only when DVI signal is sent:

Value	Meaning
0	Single Link
1	Dual Link

- **HdcpActive**

  The property is "true" if the signal is HDCP encrypted.

- **Resolution**

  The resolution of the incoming video signal (active area).

- **TotalSize**

  The resolution of the incoming video signal with the blanking area.

- **PixelClock**

  The frequency of the pixel clock.

- **3dType**

  The property shows the 3D type of the incoming signal:

Value	Meaning
0	2D signal is present
1	Frame packing
2	Field alternative
3	Line alternative
4	Side-by-side
5	L+depth

- **ColorDepth**

  The number of bit per pixel, values can be:

Value	Meaning
1	10 bit per pixel
2	12 bit per pixel
3	16 bit per pixel

- **ColorSpace**

  The colorspace of the incoming video signal:

Value	Meaning
1	RGB
2	YCbCr 4:4:4
3	YCbCr 4:2:2

- **Scan**

  The scanning method of the incoming video signal:

Value	Meaning
0	Progressive
1	Interlaced

- **ColorRange**

  Color range parameter: expand or compress.

- **OriginalColorDepth**

  The original number of bit per pixel of the incoming video signal:

Value	Meaning
1	10 bit per pixel
2	12 bit per pixel
3	16 bit per pixel

- **OriginalColorSpace**

  The original colorspace of the incoming video signal:

Value	Meaning
1	RGB
2	YCbCr 4:4:4
3	YCbCr 4:2:2

- **OriginalColorRange**

  The original value of the incoming video signal's color range.

- **AviIF**

  The AVI info frame.

- **VsIF**

  The vendor specific info frame.

- **EmbeddedAudioPresent**

  Shows if the incoming video signal contains embedded audio or not:

Value	Meaning
0	No
1	Yes
F	Unknown

- **HsyncInfo**

  Describes the timing of Hsync signal.

- **VsyncInfo**

  Describes the timing of Vsync signal.

- **PortId**

  Port identification number – not in use.

- **PortDirection**

The signal direction of the port as follows:

Value	Meaning
0	Input
1	Output
2	In-Out

- **ActiveSublayerMask**

  Active Sublayer Mask – not in use.

- **Text**

  Displayed text of the port. The allowed length is between 0-15 characters.

- **InputHdcpEnable**

  Sets the HDCP capability of the input port. "True" indicates that the port is HDCP-capable, "false" means port is not HDCP-capable.

- **Power5vIn**

  The value is "true" if 5V is detected on the connector.

- **Refresh**

  Updates the hsync and vsync info properties.

- **FactoryDefaults**

  Restore factory default settings in the given node.

5.8.3. Embedded Audio node

Path: /ROOM/<name_of_room>/PORTS/VIDEO/I<port_number>/EMBEDDEDAUDIO

Example: /ROOM/ROUTER/PORTS/VIDEO/I65/EMBEDDEDAUDIO

This node contains the embedded audio parameters of the video stream of the given input port.

- **SpdifMode**

  The active mode of the S/PDIF setting:

Value	Meaning
0 / E	embed audio
1 / D	extract audio (deembed)

- **EmbedSource**

  Source of the embedded audio:

Value	Meaning
0 / H	original HDMI stream
1 / S	S/PDIF
2 / I	I^2S

- **HdmiAudioOutputEn**

  Enable or disable output embedded audio:

Value	Meaning
true	output embedded audio is enabled
false	output embedded audio is disabled

- **Connected**

  Cable connections status:

Value	Meaning
0	no cable is present
1	cable is present
F	unknown

- **SignalPresent**

  True, if there is a valid signal on the audio port:

Value	Meaning
0	no valid signal on the audio port
1	valid signal on the audio port

- **SignalType**

  The type of the audio signal:

Value	Meaning
0	PCM
1	IEC 61937
2	DSD
3	High bitrate

- **Layout**

  The packet layout of the audio signal:

Value	Meaning
0	2 channels
1	up to 8 channels

- **SamplingFreq**

  The sampling frequency of the audio signal:

Value	Meaning
0	44.1 kHz
1	Undefined
2	48 kHz
3	32 kHz
4	22.05 kHz
6	24 kHz
8	88.2 kHz
9	768 kHz
A	96 kHz
C	176.4 kHz
E	192 kHz

- **ChannelStatus**

  <To be defined>

- **AudioIf**

  The audio info frame.

- **PortId**

  Port identification number – not in use.

- **PortDirection**

  Port direction as follows:

Value	Meaning
0	Input
1	Output
2	In-Out

- **ActiveSublayerMask**

  Active Sublayer Mask – not in use.

- **Text**

  Displayed text of the port. The allowed length is between 0-15 characters.

- **FactoryDefaults**

  Restore factory default settings in the given node.

5.9. LW2-compatibility

Lightware 2 protocol has been used by many Lightware matrices and was developed for simple switching. Because of many third party controller supports LW2, the 25G routers can be configured to use LW2 for compatibility reasons, however due to the limits of the protocol capabilities, only a subset of functions can be accessed by using this protocol. The protocol is able to switching in a single predefined room on the different layers and performing the destination mute/unmute and destination lock/unlock commands.

Connection

RS232 and TCP/IP connections are also available for 25G frames. The LW2 protocol access mode must be configured from the built-in touch panel. As the router can handle various number of virtual rooms, the user must specify the settings for each controllers separately.

Basic syntax

The matrices accept commands surrounded by curly brackets - { } - and responds data surrounded by round brackets - ( ) - only if a command was successfully executed. All input commands are converted to uppercase, but respond commands can contain upper and lower case letters as well.

Legend for control commands:

<in> = input number in 1 or 2 digit ASCII format (01,5,07,16 etc.)

<out> = output number in 1 or 2 digit ASCII format

<in²> = input number in 2 digit ASCII format (01, 02, 10, 12 etc.)

<out²> = output number in 2 digit ASCII format (01, 02, 10, 12 etc.)

<loc> = location number in 1, 2 or 3 digit ASCII format

CrLf = Carriage return, Line feed (0x0D, 0x0A)

● = space character (0x20)

→ = each command issued by the controller

← = each response received from the router

5.10. LW2 status commands

5.10.1. View product type

Description: The router responds its name.

Format	Example
Command {i} Response (<PRODUCT_TYPE>)CrLf	→ {i} ← (I: 25G-FR160)CrLf

5.10.2. View serial number

Description: The router responds its 8-digit serial number.

Format	Example
Command {s} Response (SN:<SERIAL_N>)CrLf	→ {s} ← (SN:11270142)CrLf

Info: Only the last 4 numbers are written onto the back of the device.

5.11. LW2 control commands

The following commands with <A/V/R/I/U> option can take effect in multiple layers, according to their parameters. For example, 'A' or 'V' it can change only the Audio, or only the Video layer. Further layers are referred as 'R' (RS232 layer), 'I' (Infra layer) and 'U' (USB KVM layer). Multiple layers can be selected by using multiple layer selector characters. E.g. 'ARI' refers to the Audio, RS232 and Infra layer.

As the LW2 protocol can control only one virtual room, the size of the rooms (meaning the number of inputs and number of outputs) can varies from 1x1 to 160x160. The port numbers in a room are independent from the physical port numbers, the real physical layout must be configured from the front touch panel.

Info: <A/V/R/I/U> option usually can be skipped for legacy purposes.

In this case using router commands the router changes all layers, but using status commands it displays information about only the Video layer.

5.11.1. Switch one input to one output

Description: This command switches an input to an output on the selected layers.

Format	Example
Command {<in>@<out>●<A/V/R/I/U>}	→ {4@1●AV}
Response (O<out²>●I<in²>●<A/V/R/I/U>)CrLf	← (O01●I04●AV)CrLf

Legend: <A/V/R/I/U>: Layers

Explanation: Audio and Video from input 4 is connected to output 1.

Info: If the command is used without the <A/V/R/I/U> parameter, all layers are switched.

5.11.2. Switch one input to all outputs

Description: This command switches an input to all outputs on the selected layers.

Format	Example
Command {<in>@O●<A/V/R/I/U>}	→ {3@O●AV}
Response (I<in²>●ALL●<A/V/R/I/U>) CrLf	← (I03●ALL●AV)CrLf

Legend: <A/V/R/I/U>: Layers

Explanation: The example shows how to connect all outputs to input 3.

Info: If the command is used without the <A/V/R/I/U> parameter, all layers are switched.

5.11.3. View connection on an output

Description: This command shows the **video** connection status of an output.

Format		Example
Command	{<out>?}	→ {?2}
Response	(O<out²>●I<in²>)CrLf	← (O02●I03)CrLf

Explanation: The example shows that **video** output 2 is connected to input 3.

Info: This command kept for legacy purposes; to get information about all layers, please use the multilayer command, see section 5.11.4 on page 107.)

Info: The response shows connections only for the video layer.

5.11.4. View connection on all outputs

Description: This command displays all connections on a single or multiple layers.

Format		Example for a 4x4 room
Command	{VC●<*A/V/R/I/U*>}	→ {VC●AV}
Response	(ALLV●<in²>…<in²>)CrLf	← (ALLV●02●02●02●03)CrLf
	(ALLA●<in²>…<in²>)CrLf	(ALLA●02●02●02●04)CrLf
	(ALLR●<in²>…<in²>)CrLf	(ALLR●02●02●02●04)CrLf
	(ALLI●<in²>…<in²>)CrLf	(ALLI●02●02●02●04)CrLf
	(ALLU●<in²>…<in²>)CrLf	(ALLU●02●02●02●04)CrLf

Legend: <A/V/R/I/U>: Layers

<in²>: Contains mute & lock state of the output:

M: muted

L: locked

U: muted & locked.

and contains the number of the selected input

(02: connected to input no. 2)

(M02: muted output, selected input: no. 2)

(L02: locked to input: no. 2)

(U02: muted & locked output, selected input: no. 2)

Explanation: The response contains all the connections, if both channels are selected the response is two messages. (ALLV●M02●U02●L02●03) response means video output 1 is muted, video output 2 is muted & locked and video output 3 is locked to input 2.

Info: If the command is used without the <A/V/R/I/U> parameter, the response shows only the video layer connections.

5.11.5. Mute specified output

Description: This command mutes an output on a single or multiple layers.

Format		Example
Command	{#<out>●<A/V/R/I/U>}	→ {#2●A}
Response	(1MT<out²>●<A/V/R/I/U>) CrLf	← (1MT02●A)CrLf

Legend: <A/V/R/I/U>: Layers

Explanation: The example shows how to mute audio output no. 2.

Info: If the command is used without the <A/V/R/I/U> parameter, all layers are muted.

5.11.6. Unmute specified output

Description: This command mutes an output on a single or multiple layers.

Format		Example
Command	{+<out>●<A/V/R/I/U>}	→ {+2●A}
Response	(0MT<out²>●<A/V/R/I/U>) CrLf	← (0MT02●A)CrLf

Legend: <A/V/R/I/U>: Layers

Explanation: The example shows how to unmute audio output no. 2.

5.11.7. Lock specified output

Description: This command locks an output on a single or multiple layers.

Format		Example
Command	{#><out>●<A/V/R/I/U>}	→ {#>4●AV}
Response	(1LO<out²>●<A/V/R/I/U>) CrLf	← (1LO04●AV)CrLf

Legend: <A/V/R/I/U>: Layers

Explanation: The example shows how to lock audio & video on output no. 4.

5.11.8. Unlock specified output

Description: This command unlocks an output on a single or multiple layers.

Format		Example
Command	{+<<out>●<A/V/R/I/U>}	→ {+<4●AV}
Response	(0LO<out²>●<A/V/R/I/U>) CrLf	← (0LO04●AV)CrLf

Legend: <A/V/R/I/U>: Layers

Explanation: The example shows how to unlock audio & video on output no. 4.

6. Specifications

Media layers

Video data rate ..... 25 Gbit/sec per port
Video compatibility ..... DisplayPort 1.2, HDMI 2.0 with 3D,
..... Single- and Dual-Link DVI, 3G-SDI
Audio ..... 3 layers – Embedded-, Forward- and Return audio channels
Audio compatibility .....S/PDIF 7.1, 5.1 Dolby Digital, DTS Audio,
.....HDMI 1.4 Embedded audio (with ARC), stereo and multichannel PCM
Ethernet ..... 100 Mbit/port (total 320) with 1 Gigabit uplink
USB KVM ..... USB HID crosspoint and extension
RS232 and IR ..... Control for all devices through the matrix
CEC.....According to the HDMI standard

Control

Ethernet ..... Redundant control (one for each CPU)
Ethernet control..... Ethernet 10Base-T or 100base-TX (Auto-negotiation)
RS232 ..... Redundant control (one for each CPU)
RS-232 Baud rate ..... Selectable baud rate (9600-115200, default: 57600)
Room and user management ..... Unlimited rooms and users
Virtual matrix option ..... Virtual I/O numbering, Virtual matrices
3rd party control ..... Vista Spyder and Barco Encore compatible

Connectors (frame)

Ethernet control..... 2 x RJ45 (1-1 per CPU)
RS-232 control ..... 2 x 9 pole D-sub (1-1 per CPU)
Ethernet layer..... 2x RJ45 – 1 Gigabit uplink for Ethernet
SMPTE 169M Alarm output .....1 x BNC
Power ..... 4 x IEC-320 C-20

General

Crosspoint size ..... From 8 x 8 up to 160 x 160
Power ..... 100-240 V AC
Power consumption ..... 300 W (typ) – without I/O boards
Power consumption .....2000 W (typ)* – with I/O boards
Enclosure dimensions..... 446 (482) W x 640 D x 1866 H mm
Height in rack units 42U
Net weight .....200 kg (without I/O boards)
Temperature ..... 0°C to 50°C operational, -40°C to +70°C storage
Humidity ..... 10 to 90% non-condensing
RoHS compliance ..... Yes

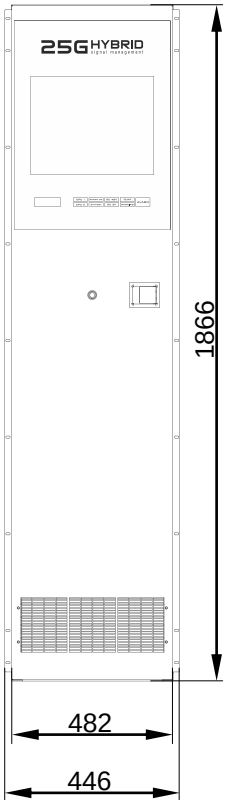
* Depends on the current configuration.

Redundancy & Reliability

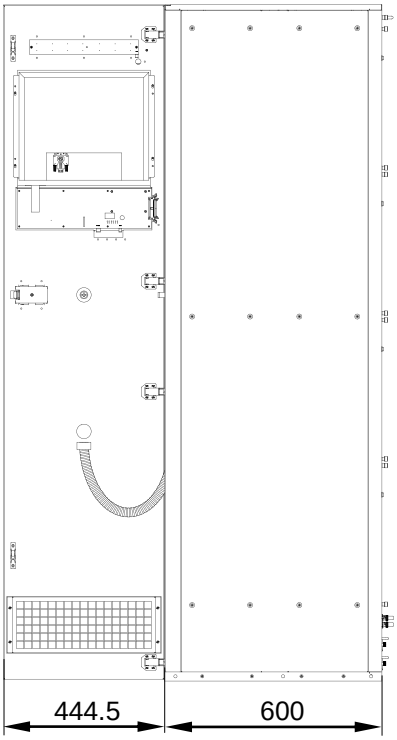
CPU	Dual redundant
Hot swappable	Each IO board / CPU / fan tray / PSU
Power supplies	Maximum 6 PSUs
PSU redundancy	Up to N+2
MTBF	30.000 hours

6.1. Technical drawings

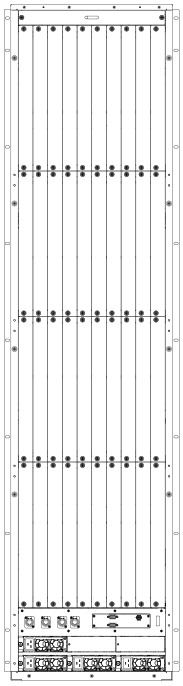
Front view



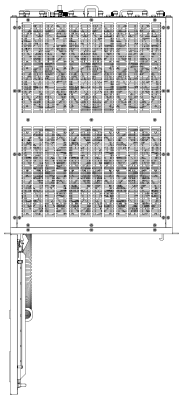
Side view (front panel open)



Rear view



Top view



7. Warranty

Lightware Visual Engineering warrants this product against defects in materials and workmanship for a period of three years from the date of purchase.

The customer shall pay shipping charges when unit is returned for repair. Lightware will cover shipping charges for return shipments to customers.

In case of defect please contact your local representative, or Lightware at

Lightware Visual Engineering

H-1071 Budapest, Peterdy Street 15, HUNGARY

E-mail: support@lightware.eu

8. Document revision history

Document	Release Date	Changes	Editor
Rev. 1.0	06-12-2013	Initial version	Laszlo Zsedenyi
Rev. 1.1	06-06-2014	LW3 programmers' reference is updated	Laszlo Zsedenyi
Rev. 1.2	01-09-2014	FR160 changes, Control Software update, Programmers' reference improved	Laszlo Zsedenyi